Activity Report 2017

Project-Team INDES

Secure Diffuse Programming

RESEARCH CENTER
Sophia Antipolis - Méditerranée

THEME
Distributed programming and Software engineering

Table of contents

Project-Team INDES

Creation of the Team: 2009 January 01, updated into Project-Team: 2010 July 01

Keywords:

Computer Science and Digital Science:

A1.3. - Distributed Systems
A2. - Software
A2.1. - Programming Languages
A2.1.3. - Functional programming
A2.1.7. - Distributed programming
A2.1.8. - Synchronous languages
A2.1.9. - Dynamic languages
A2.2.1. - Static analysis
A2.2.3. - Run-time systems
A4. - Security and privacy
A4.3.3. - Cryptographic protocols
A4.6. - Authentication
A4.7. - Access control
A4.8. - Privacy-enhancing technologies

Other Research Topics and Application Domains:

B6.3.1. - Web
B6.4. - Internet of things
B9.4.1. - Computer science
B9.8. - Privacy

1. Personnel

Research Scientists

Manuel Serrano [Team leader, Inria, Senior Researcher, HDR]
Nataliia Bielova [Inria, Researcher]
Ilaria Castellani [Inria, Researcher]
Tamara Rezk [Inria, Researcher]
Bernard Serpette [Inria, Researcher, until Feb 2017]

Post-Doctoral Fellow

Nguyen Nhat Minh Ngo [Inria]

PhD Students

Dolière Francis Somé [Inria]
Colin Vidal [Inria]

Technical staff

Cédric Duminy [Inria, until Mar 2017]
Vincent Prunet [Inria, until Mar 2017]

Interns

Imane Fouad [Inria, from Mar 2017 until Aug 2017 and from Oct 2017 until Dec 2017]
Mohamad El Laz [Inria, from Apr 2017 until Aug 2017]

Administrative Assistant
Nathalie Bellesso [Inria]

Visiting Scientists
Reynald Affeldt [Nov 2017]
Marc Feeley [University of Montréal, from May 2017 until Jul 2017]
Bertrand Petit [from Sep 2017]

2. Overall Objectives

2.1. Overall Objectives

The goal of the Indes team is to study models for diffuse computing and develop languages for secure diffuse applications. Diffuse applications, of which Web 2.0 applications are a notable example, are the new applications emerging from the convergence of broad network accessibility, rich personal digital environment, and vast sources of information. Strong security guarantees are required for these applications, which intrinsically rely on sharing private information over networks of mutually distrustful nodes connected by unreliable media.

Diffuse computing requires an original combination of nearly all previous computing paradigms, ranging from classical sequential computing to parallel and concurrent computing in both their synchronous / reactive and asynchronous variants. It also benefits from the recent advances in mobile computing, since devices involved in diffuse applications are often mobile or portable.

The Indes team contributes to the whole chain of research on models and languages for diffuse computing, going from the study of foundational models and formal semantics to the design and implementation of new languages to be put to work on concrete applications. Emphasis is placed on correct-by-construction mechanisms to guarantee correct, efficient and secure implementation of high-level programs. The research is partly inspired by and built around Hop, the web programming model proposed by the former Mimosa team, which takes the web as its execution platform and targets interactive and multimedia applications.

3. Research Program

3.1. Parallelism, concurrency, and distribution

Concurrency management is at the heart of diffuse programming. Since the execution platforms are highly heterogeneous, many different concurrency principles and models may be involved. Asynchronous concurrency is the basis of shared-memory process handling within multiprocessor or multicore computers, of direct or fifo-based message passing in distributed networks, and of fifo- or interrupt-based event handling in web-based human-machine interaction or sensor handling. Synchronous or quasi-synchronous concurrency is the basis of signal processing, of real-time control, and of safety-critical information acquisition and display. Interfacing existing devices based on these different concurrency principles within HOP or other diffuse programming languages will require better understanding of the underlying concurrency models and of the way they can nicely cooperate, a currently ill-resolved problem.

3.2. Web and functional programming

We are studying new paradigms for programming Web applications that rely on multi-tier functional programming. We have created a Web programming environment named HOP. It relies on a single formalism for programming the server-side and the client-side of the applications as well as for configuring the execution engine.

HOP is a functional language based on the SCHEME programming language. That is, it is a strict functional language, fully polymorphic, supporting side effects, and dynamically type-checked. HOP is implemented as an extension of the BIGLOO compiler that we develop. In the past, we have extensively studied static analyses (type systems and inference, abstract interpretations, as well as classical compiler optimizations) to improve the efficiency of compilation in both space and time.

3.3. Security of diffuse programs

The main goal of our security research is to provide scalable and rigorous language-based techniques that can be integrated into multi-tier compilers to enforce the security of diffuse programs. Research on language-based security has been carried on before in former Inria teams. In particular previous research has focused on controlling information flow to ensure confidentiality.

Typical language-based solutions to these problems are founded on static analysis, logics, provable cryptography, and compilers that generate correct code by construction. Relying on the multi-tier programming language HOP that tames the complexity of writing and analysing secure diffuse applications, we are studying language-based solutions to prominent web security problems such as code injection and cross-site scripting, to name a few.

4. New Software and Platforms

4.1. Bigloo

KEYWORD: Compilers
FUNCTIONAL DESCRIPTION: Bigloo is a Scheme implementation devoted to one goal: enabling Scheme based programming style where C(++) is usually required. Bigloo attempts to make Scheme practical by offering features usually presented by traditional programming languages but not offered by Scheme and functional programming. Bigloo compiles Scheme modules. It delivers small and fast stand alone binary executables. Bigloo enables full connections between Scheme and C programs, between Scheme and Java programs.

RELEASE FUNCTIONAL DESCRIPTION: modification of the object system (language design and implementation), new APIs (alsa, flac, mpg123, avahi, csv parsing), new library functions (UDP support), new regular expressions support, new garbage collector (Boehm's collection 7.3alpha1).

- Participant: Manuel Serrano
- Contact: Manuel Serrano
- URL: http://www-sop.inria.fr/teams/indes/fp/Bigloo/

4.2. Camloo

KEYWORD: Compilers
FUNCTIONAL DESCRIPTION: Camloo is a caml-light to bigloo compiler, which was developed a few years ago to target bigloo 1.6c. New major releases 0.4.x of camloo have been done to support bigloo 3.4 and bigloo 3.5. Camloo makes it possible for the user to develop seamlessly a multi-language project, where some files are written in caml-light, in C, and in bigloo. Unlike the previous versions of camloo, 0.4.x versions do not need a modified bigloo compiler to obtain good performance. Currently, the only supported backend for camloo is bigloo/C. We are currently rewriting the runtime of camloo in bigloo to get more portability and to be able to use HOP and camloo together.

- Contact: Manuel Serrano

4.3. Hop

KEYWORDS: Programming language - Multimedia - Iot - Web 2.0 - Functional programming

SCIENTIFIC DESCRIPTION: The Hop programming environment consists in a web broker that intuitively combines in a single architecture a web server and a web proxy. The broker embeds a Hop interpreter for executing server-side code and a Hop client-side compiler for generating the code that will get executed by the client.

An important effort is devoted to providing Hop with a realistic and efficient implementation. The Hop implementation is validated against web applications that are used on a daily-basis. In particular, we have developed Hop applications for authoring and projecting slides, editing calendars, reading RSS streams, or managing blogs.
FUNCTIONAL DESCRIPTION: Multitier web programming language and runtime environment.

- Participant: Manuel Serrano
- Contact: Manuel Serrano
- URL: http://hop.inria.fr

4.4. IFJS

Infomation Flow monitor inlining for JavaScript
FUNCTIONAL DESCRIPTION: The IFJS compiler is applied to JavaScript code. The compiler generates JavaScript code instrumented with checks to secure code. The compiler takes into account special features of JavaScript such as implicit type coercions and programs that actively try to bypass the inlined enforcement mechanisms. The compiler guarantees that third-party programs cannot (1) access the compiler internal state by randomizing the names of the resources through which it is accessed and (2) change the behaviour of native functions that are used by the enforcement mechanisms inlined in the compiled code.

- Contact: Manuel Serrano
- URL: http://www-sop.inria.fr/indes/ifJS/

4.5. iflowsigs.js

KEYWORDS: Compilers - Monitoring
FUNCTIONAL DESCRIPTION: iflowsigs.js is a JavaScript library designed to inline an information flow monitor into JavaScript code. iflowsigs.js support is able to track information flow even in programs that interact with arbitrary Web APIs.

- Participants: José Fragoso Santos and Tamara Rezk
- Contact: Tamara Rezk
- URL: http://j3fsantos.github.io/PersonalPage/IFMonitor/

4.6. iflowTYPES.js

FUNCTIONAL DESCRIPTION: iflowtypes.js is a JavaScript library designed to type secure information flow in JavaScript. iflowtypes.js has two main modes of operation: fully static and hybrid. In the hybrid mode, the program to be typed is instrumented with runtime assertions that are verified at runtime. By deferring rejection to runtime, the hybrid type system is able to type more programs than fully static mechanisms.

- Contact: Tamara Rezk
- URL: http://j3fsantos.github.io/PersonalPage/TypeSystem/

4.7. Mashic

FUNCTIONAL DESCRIPTION: The Mashic compiler is applied to mashups with untrusted scripts. The compiler generates mashups with sandboxed scripts, secured by the same origin policy of the browsers. The compiler is written in Bigloo.

- Contact: Manuel Serrano
- URL: http://web.ist.utl.pt/~ana.matos/Mashic/mashic.html

4.8. scheme2JS

Scheme to JavaScript
KEYWORD: Compilers
FUNCTIONAL DESCRIPTION: Scm2JS is a Scheme to JavaScript compiler distributed under the GPL license. Even though much effort has been spent on being as close as possible to R5rs, we concentrated mainly on efficiency and interoperability. Usually Scm2JS produces JavaScript code that is comparable (in speed) to hand-written code. In order to achieve this performance, Scm2JS is not completely R5rs compliant. In particular it lacks exact numbers.

Interoperability with existing JavaScript code is ensured by a JavaScript-like dot-notation to access JavaScript objects and by a flexible symbol-resolution implementation.

Scm2JS is used on a daily basis within Hop, where it generates the code which is sent to the clients (web-browsers).

- Contact: Manuel Serrano
- URL: http://www-sop.inria.fr/indes/scheme2js/

4.9. Hiphop.js

KEYWORDS: Web 2.0 - Synchronous Language
FUNCTIONAL DESCRIPTION: Hiphop.js is a DSL which extends JavaScript with temporal constructions. It makes easier the orchestration of asynchronous Web applications.

- Contact: Colin Vidal
- URL: http://www-sop.inria.fr/members/Colin.Vidal/hiphop/

4.10. Server-Side Protection against Third Party Web Tracking

KEYWORDS: Privacy - Web Application - Web - Architecture - Security by design - Program rewriting techniques
FUNCTIONAL DESCRIPTION: We present a new web application architecture that allows web developers to gain control over certain types of third party content. In the traditional web application architecture, a web application developer has no control over third party content. This allows the exchange of tracking information between the browser and the third party content provider.

To prevent this, our solution is based on the automatic rewriting of the web application in such a way that the third party requests are redirected to a trusted third party server, called the Middle Party Server. It may be either controlled by a trusted party, or by a main site owner and automatically eliminates third-party tracking cookies and other technologies that may be exchanged by the browser and third party server

- Contact: Doliere Some
- URL: http://www-sop.inria.fr/members/Doliere.Some/essos/

4.11. BELL

Browser fingerprinting via Extensions and Login-Leaks
KEYWORDS: Browser Extensions - Security and Privacy in Web Services - Social Networks Security and Privacy
FUNCTIONAL DESCRIPTION: Recent studies show that users can be tracked based on their web browser properties. This software is designed to conduct an experiment on such kinds of user tracking. In this experiment, we demonstrate that a Web user can also be tracked by

- her browser extensions (such as AdBlock, Pinterest, or Ghostery), and

- the websites she has logged in (such as Facebook, Gmail, or Twitter).

In the experiment, we collect user's browser fingerprint, together with the browser extensions installed and a list of websites she has logged in. We only collect anonymous data during the experiment (more details in our Privacy Policy [1]), we will securely store the data on an Inria server, use it only for research purposes and not share it with anyone outside of Inria.

- Contact: Gabor Gulyas
- URL: https://extensions.inrialpes.fr/

4.12. webstats

Webstats
KEYWORDS: Web Usage Mining - Statistic analysis - Security
FUNCTIONAL DESCRIPTION: The goal of this tool is to perform a large-scale monthly crawl of the top Alexa sites, collecting both inline scripts (written by web developers) and remote scripts, and establishing the popularity of remote scripts (such as Google Analytics and jQuery). With this data, we establish whether the collected scripts are actually written in a subset of JavaScript by analyzing the different constructs used in those scripts. Finally, we collect and analyze the HTTP headers of the different sites visited, and provide statistics about the usage of HTTPOnly and Secure cookies, and the Content Security Policy in top sites.

- Contact: Doliere Some
- URL: https://webstats.inria.fr

5. New Results

5.1. Type Abstraction for Relaxed Noninterference

Information-flow security typing statically prevents confidential information to leak to public channels. The fundamental information flow property, known as *noninterference*, states that a public observer cannot learn anything from private data. As attractive as it is from a theoretical viewpoint, noninterference is impractical: real systems need to intentionally declassify some information, selectively. Among the different information flow approaches to declassification, a particularly expressive approach was proposed by Li and Zdancewic, enforcing a notion of *relaxed noninterference* by allowing programmers to specify *declassification policies* that capture the intended manner in which public information can be computed from private data. The paper [15] shows how we can exploit the familiar notion of type abstraction to support expressive declassification policies in a simpler, yet more expressive manner. In particular, the type-based approach to declassification—which we develop in an object-oriented setting—addresses several issues and challenges with respect to prior work, including a simple notion of label ordering based on subtyping, support for recursive declassification policies, and a local, modular reasoning principle for relaxed noninterference. This work paves the way for integrating declassification policies in practical security-typed languages.

5.2. Multiparty Reactive Sessions

Synchronous reactive programming (SRP) is a well-established programming paradigm whose essential features are logical instants, broadcast events and event-based preemption. This makes it an ideal vehicle for the specification and analysis of reactive systems, and indeed several programming languages and frameworks based on SRP have been put forward. On the other hand, *session-based concurrency* is the model of concurrent computation induced by *session types*, a rich typing discipline designed to specify the structure of interactions. In a nutshell, session types describe communication protocols between two or more participants by specifying the sequencing of messages along communication channels, as well as their functionality (sender, receiver and type of carried data). Originally conceived as a static analysis technique for an enhanced version of the π-calculus, session types have been subsequently transferred to functional, concurrent, and object-oriented programming languages, and adapted to support run-time verification.

[1] https://extensions.inrialpes.fr/privacy.php

A combination of session-based concurrency and SRP features appears to be appropriate to specify and analyse communication-centric systems in which some components may have a reactive and/or timed behaviour. In joint work with colleagues from I3S and the University of Groningen, currently submitted, we study the integration of SRP and session-based concurrency. To this end, we propose a calculus for multiparty sessions enriched with features from SRP. In this calculus, protocol participants communicate by broadcast messages, have the ability to suspend themselves while waiting for an absent message, and may react to the presence of particular events by triggering alternative behaviours. We equip the calculus with a session type system which enforces expected session properties such as communication safety, protocol fidelity, and input lock freedom. This session type system departs significantly from existing ones: the interplay of classical, well-established assumptions of SRP with session-based constructs requires revisiting central notions of multiparty session types, such as those of global type, local type and projection.

5.3. Multiparty Reversible Sessions

Reversibility has been an active trend of research for the last fifteen years. A reversible computation is a computation that has the ability to roll back to a past state. Allowing computations to reverse is a means to improve system flexibility and reliability. In the setting of concurrent process calculi, reversible computations have been first studied for CCS, then for the π-calculus, and only recently for session calculi. In [14] we present a multiparty session calculus with reversible computations. Our proposal improves on existing reversible session calculi in several respects: it allows for concurrent and sequential composition within processes and types, it gives a compact representation of the *past* of processes and types, which facilitates the definition of rollback, and it implements a fine-tuned strategy for backward computation. We propose a refined session type system for this calculus and show that it enforces the expected properties of session fidelity, forward and backward progress, as well as causal consistency. In conclusion, our calculus is a conservative extension of previous proposals, offering enhanced expressive power and refined analysis techniques.

5.4. JavaScript ahead-of-time compilation

Nowadays, JavaScript is no longer confined to the programming of web pages. It is also used for programming server-side parts of web applications, compilers, and there is a growing trend for using it for programming internet-of-things (IoT) applications. All major industrial actors of the field are looking for, or are already providing, JavaScript based development kits (IoT.js, Espruino, JerryScript, Kinoma.js, ...). In this application domain, JavaScript programs execute on tiny devices that have limited hardware capacities, for instance only a few kilobytes of memory. Just-in-time (JIT) compilation, which has proved to be so effective for improving JavaScript performances, is unthinkable in these constrained environments. There would be just not enough memory nor CPU capacity to execute them at runtime. Pure JavaScript interpreters are then used but this comes with a strong performance penalty, especially when compared to assembly or C programs, that limits the possible uses.

When JIT compilation is not an option and when interpretation is too slow, the alternative is static compilation, also known as ahead-of-time (AOT) compilation. It has the promise of combining small memory footprints and good performances. However, this implementation technique seems not to fit the JavaScript design whose unique combination of antagonistic features such as functional programming support, high mutation rates of applications, introspection, and dynamicity, makes most known classical AOT compilation techniques ineffective.

Indeed, JavaScript is hard to compile, much harder than languages such as C, Java, and even harder than other functional languages like Scheme and ML. This is because a JavaScript source code accepts many more possible interpretations than other languages do. It forces JavaScript compilers to adopt a defensive position by generating target codes that can cope with all the possible, even unlikely, interpretations. This difficulty probably explains why JavaScript AOT compilation has received so little attention from the scientific community. All these difficulties cannot be solved with traditional compilation techniques. They demand new strategies. This is what we explore. We are developing a prototype of a new compiler that distinguishes from classical compilers by relying on static program analyses that are not governed by approximating all

possible program executions but by inferring properties that suit the compiler back-end. For instance, instead of inferring types that describe a super set of all possible executions, this compiler infers types for which the compiler is able to deliver good code.

The whole year has been devoted to implementing an operational prototype of the compiler. The preliminary results we have obtained are very promising but we still have to improve the code generation quality before writing and publishing complete reports describing it. This will be one of our main objectives for 2018.

5.5. Orchestration of Web applications

Modern Web applications are composed of numerous heterogeneous actors (users, distant servers and services, IoT devices, etc.) interacting together by means of asynchronous events. The harmonious interaction between these actors is called *orchestration*. JavaScript, the mainstream language for writing Web applications, enables programmers to orchestrate events with an asynchronous event-loop. However, event-loop based orchestration is known to be a difficult problem leading to programs which are difficult to write, read and maintain. To address this problem, Hiphop.js, a domain-specific language (DSL), has been developed during the last two years. It extends JavaScript by means of *temporal constructors* allowing explicit synchronization, parallelism and preemption. These constructors are inspired from the Esterel synchronous language.

During this year Hiphop.js has gained in maturity. First, a development environment has been developed. It is now possible to debug Hiphop.js programs by visualizing the code source during the execution, inspecting instructions state and signals value. Moreover, it is possible to queue reactions in order to analyze step-by-step the global state of the program between each reaction. The debugger can also be used and controlled remotely, using a simple Web browser. It is an important feature since Hiphop.js applications can run on different types of devices, especially smartphones or headless devices, on which debugging is impossible. Besides, in order to have a deeper integration with JavaScript and to make the adoption of Hiphop.js easier for new users, a new syntax has been designed and implemented.

A short paper describing HipHop has been accepted for publication at the SAC'18 symposium.

Finally, Hiphop.js is used in the context of a music show during the *MANCA* (http://www.cirm-manca.org/manca2017/) festival in Nice. It is used to orchestrate the composition of lights and songs during the show. Moreover, the public can interact with musicians by the means of smartphones, playing specific songs during delimited periods of the performance. Those interactions are implemented using Hiphop.js.

5.6. On the Content Security Policy Violations due to the Same-Origin Policy

Modern browsers implement different security policies such as the Content Security Policy (CSP), a mechanism designed to mitigate popular web vulnerabilities, and the Same Origin Policy (SOP), a mechanism that governs interactions between resources of web pages.

In the work [17], we describe how CSP may be violated due to the SOP when a page contains an embedded iframe from the same origin. We analyse 1 million pages from 10,000 top Alexa sites and report that at least 31.1% of current CSP-enabled pages are potentially vulnerable to CSP violations. Further considering real-world situations where those pages are involved in same-origin nested browsing contexts, we found that in at least 23.5% of the cases, CSP violations are possible.

During our study, we also identified a divergence among browsers implementations in the enforcement of CSP in srcdoc sandboxed iframes, which actually reveals a problem in Gecko-based browsers CSP implementation. To ameliorate the problematic conflicts of the security mechanisms, we discuss measures to avoid CSP violations.

5.7. Control What You Include! Server-Side Protection Against Third Party Web Tracking

Third party tracking is the practice by which third parties recognize users accross different websites as they browse the web. Recent studies show that 90% of websites contain third party content that is tracking its

users across the web. Website developers often need to include third party content in order to provide basic functionality. However, when a developer includes a third party content, she cannot know whether the third party contains tracking mechanisms. If a website developer wants to protect her users from being tracked, the only solution is to exclude any third-party content, thus trading functionality for privacy.

We describe and implement a privacy-preserving web architecture [16] that gives website developers a control over third party tracking: developers are able to include functionally useful third party content, and at the same time ensuring that the end users are not tracked by the third parties.

5.8. A Better Facet of Dynamic Information Flow Control

Multiple Facets (MF) is a dynamic enforcement mechanism which has proved to be a good fit for implementing information flow security for JavaScript. It relies on multi executing the program, once per each security level or view, to achieve soundness. By looking inside programs, MF encodes the views to reduce the number of needed multi-executions.

We extend Multiple Facets in three directions. First, we propose a new version of MF for arbitrary lattices, called Generalised Multiple Facets, or GMF. GMF strictly generalizes MF, which was originally proposed for a specific lattice of principals. Second, we propose a new optimization on top of GMF that further reduces the number of executions. Third, we strengthen the security guarantees provided by Multiple Facets by proposing a termination sensitive version that eliminates covert channels due to termination.

5.9. Impossibility of Precise and Sound Termination Sensitive Security Enforcements

An information flow policy is termination sensitive if it imposes that the termination behaviour of programs is not influenced by confidential input. Termination sensitivity can be statically or dynamically enforced. On one hand, existing static enforcement mechanisms for termination sensitive policies are typically quite conservative and impose strong constraints on programs like absence of while loops whose guard depends on confidential information. On the other hand, dynamic mechanisms can enforce termination sensitive policies in a less conservative way. SME, one of such mechanisms, was even claimed to be sound and precise in the sense that the enforcement mechanism will not modify the observable behaviour of programs that comply with the termination sensitive policy. However, termination sensitivity is a subtle policy, that has been formalized in different ways. A key aspect is whether the policy talks about actual termination, or observable termination.

We prove that termination sensitive policies that talk about actual termination are not enforceable in a sound and precise way. For static enforcements, the result follows directly from a reduction of the decidability of the problem to the halting problem. However, for dynamic mechanisms the insight is more involved and requires a diagonalization argument.

In particular, our result contradicts the claim made about SME. We correct this claim by showing that SME enforces a subtly different policy that we call indirect termination sensitive noninterference and that talks about observable termination instead of actual termination. We construct a variant of SME that is sound and precise for indirect termination sensitive noninterference. Finally, we also show that static methods can be adapted to enforce indirect termination sensitive information flow policies (but obviously not precisely) by constructing a sound type system for an indirect termination sensitive policy.

5.10. BELL: Browser fingerprinting via Extensions and Login-Leaks

Recent work showed that websites can detect browser extensions that users install and websites they are logged into. This poses significant privacy risks, since extensions and Web logins can leak sensitive information and be used to track users via fingerprinting.

In joint work with Gabor Gulyas and Claude Castelluccia (Privatics team, Inria Grenoble), we report on the first large-scale study of this new form of fingerprinting, based on more than 16,000 users who visited our website [2]. Our website identifies installed Google Chrome extensions via Web Accessible Resources, and detects logged in websites by methods that rely on URL redirection and CSP violation report. Our website is able to test and detect the presence of 16,743 Chrome extensions, covering 28% of all free Chrome extensions. We also test whether the user is connected to 60 different websites.

We compute uniqueness of collected fingerprints, and find out that 54.86% of users that have installed at least one detectable extension are unique; 19.53% are unique because they logged in one or more detectable websites; and 89.23% of users are unique because they have at least one extension and one login detected.

We optimize the fingerprinting algorithm and show that it is possible to fingerprint a user in less than 625 milliseconds by selecting the most identifying combinations of extensions. Moreover, we discover that 22.98% of users can be uniquely identified and tracked by Web logins, even if they disable JavaScript. We conclude with possible countermeasures.

5.11. Large-scale measurement of invisible images for Web tracking

In joint work with Arnaud Legout (DIANA team, Inria Sophia Antipolis), we perform large scale Web measurements to evaluate Web tracking and privacy leaks in every-day Web browsing. Unlike the related work, our study focuses on the third-party HTTP requests for invisible images. We have identified two types of images that are invisible to the end user and most likely used for Web tracking: one-pixel images and empty images.

We have visited 4,351,318 pages from 38,000 web sites and identified that almost half of the third-party images are invisible to the end user. This finding raises a lot of concerns regarding Web tracking and user privacy on the Web. We made the first evaluations on how much of this invisible tracking is prevented by the popular browser extensions used for the privacy protection. We also find the top invisible trackers and the invisible trackers not blocked by the browser extensions.

We continue this work by analysing all the HTTP requests and responses that lead to invisible images in order to (1) provide a fine-grained classification of third-party cookie tracking; (2) analyse new techniques of cookie-synching used by the companies; (3) evaluate redirection chains that lead to user's information exchange between various companies; (4) identify companies that use invisible images for none of the known tracking techniques, and analyse such requests and responses further to reveal new Web tracking technologies.

6. Partnerships and Cooperations

6.1. National Initiatives

6.1.1. ANR AJACS

The AJACS project (Analyses of JavaScript Applications: Certification & Security) is funded by the ANR for 42 months, starting December 2014. The goal of the AJACS project is to provide strong security and privacy guarantees on the client side for web application scripts. The Indes members are involved in the tasks WP2 Certified Analyses and WP3 Security of JavaScript Applications. The partners of this project include Inria teams Celtique (coordinator), Toccata, and Prosecco.

6.1.2. FUI UCF

The 3 years long UCF project aims at developing a reactive Web platforms for delivering multimedia contents. The partners of the project are the startups Alterway, OCamlPro, and XWiki, and the academic research laboratories of University Pierre et Marie Curie, and Denis Diderot.

[2] https://extensions.inrialpes.fr/

6.1.2.1. Actions marquantes

Inria Sophia Antipolis Actions Marquantes is a special funding for 2 postdocs during one year to explore a new research direction. The joint project with DIANA team "User discrimination on the Web: measurement, causation and prevention" has obtained this funding. The goal of this project is to detect when users get discriminated on the Web, what are the technologies used to discriminate users and how we can prevent it without breaking the functionality and sometimes useful personalisation within Web applications.

6.2. European Initiatives

6.2.1. Collaborations in European Programs, Except FP7 & H2020

6.2.1.1. ICT Cost Action IC1405 on Reversible Computation

Program: ICT COST Action

Project title: Reversible computation - extending horizons of computing

Duration: November 2014 - November 2018

Coordinator: Irek Ulidowski, University of Leicester

Other partners: several research groups, belonging to 23 European countries.

Abstract: Reversible computation is an emerging paradigm that extends the standard mode of computation with the ability to execute in reverse. It aims to deliver novel computing devices and software, and to enhance traditional systems. The potential benefits include the design of reversible logic gates and circuits - leading to low-power computing and innovative hardware for green ICT, new conceptual frameworks and language abstractions, and software tools for reliable and recovery-oriented distributed systems. This is the first European network of excellence aimed at coordinating research on reversible computation.

6.2.1.2. Bilateral PICS project SuCCeSS

Program: CNRS PICS project

Project acronym: SuCCeSS

Project title: Security, Adaptability and time in Communication Centric Software Systems

Duration: June 2016 - June 2019

Coordinator: Cinzia Di Giusto, I3S, Sophia Antipolis

Partners: I3S, Inria, University of Groningen

Abstract: The project SuCCeSS is a CNRS-funded "Projet coopératif" (PICS 07313), involving two French teams in Sophia Antipolis (the MDSC team at the laboratory I3S, acting as coordinator, and the INDES team) and one Dutch team at the University of Groningen. The project started in June 2016 and is due to end in June 2019. The objective of the project is to study formal models for reliable distributed communication-centric software systems. The project focusses on analysis and validation techniques based on behavioural types, aimed at enforcing various properties (safety, liveness, security) of structured communications.

6.3. International Initiatives

6.3.1. Inria International Partners

6.3.1.1. Informal International Partners

Since 2009, the team has been collaborating with Mariangiola Dezani's group at the University of Torino.

6.4. International Research Visitors

6.4.1. Visits of International Scientists

In February-March, the team hosted for two weeks Professor Mariangiola Dezani-Ciancaglini from the University of Torino and Professor Paola Giannini from the University of Piemonte Orientale (Italy). The visit was partly funded by the COST Action on Reversibility.

Marc Feeley, professor at the University of Montréal has been visiting the team from April 1st to June 30th. The visit has been funded by the Labex UCN. M. Feeley has been working with M. Serrano on the compilation of functional languages in general, and JavaScript more specifically.

6.4.1.1. Internships

Web Tracking through invisible Web beacons

Imane Fouad made an internship from March 2017 until August 2017, followed by a second internship from October 2017 until December 2017. She is selected for a PhD in INDES, and will start on 1 January 2017.

Imane's internship aimed at analyzing the new Web tracking technologies based on "Web beacon", or "pixel image" tracking. This tracking technology uses an invisible 1x1 pixel image that is used to send information to third-party trackers, while being invisible to the user. Web beacon tracking is particularly invasive because it cannot be blocked by Private browsing mode, AdBlock or Ghostery extensions, and not even by disabling JavaScript.

Imane Fouad has run automated Web experiments using the OpenWPM platform and performed large-scale measurement of the Web beacon tracking on the Web. She detected which companies use Web beacon, how this technology works, and is currently analysing cookie-based tracking techniques such as redirection chains and cookie synching with the ultimate goal to provide a fine-grained classification of existing Web tracking technologies.

7. Dissemination

7.1. Promoting Scientific Activities

7.1.1. Scientific Events Organisation

7.1.1.1. General Chair, Scientific Chair

- Ilaria Castellani was the co-chair (together with Pedro D'Argenio, Mohammad Reza Mousavi and Ana Sokolova) of the workshop "Open Problems in Concurrency Theory 2017" (OPCT'17), which was held in the Institute of Science and Technology Austria (IST Austria), from June 26 to June 29. http://opct2017.famaf.unc.edu.ar/Home.html
- Ilaria Castellani was the co-chair (together with Mohammad Reza Mousavi) of the workshop TRENDS 2017, which took place in Berlin on September 9, in association with the CONCUR 2017 conference. https://concurrency-theory.org/events/workshops/trends

7.1.1.2. Member of the Organizing Committees

- Tamara Rezk was member of the Organizing Committees of the conference Euro Security and Privacy 2017 and the workshop Secure Compilation Meeting at POPL 2017. She also is a member of the Steering Committee of the POST conference.
- Manuel Serrano is a member of the Steering Committee for the Trends in Functional Programming conference and the Programming conference.
- Nataliia Bielova was a member of the Organizing Committee of the Dagstuhl Seminar on Online Privacy and Web Transparency. https://www.dagstuhl.de/en/program/calendar/semhp/?semnr=17162

7.1.2. Scientific Events Selection

7.1.2.1. Chair of Conference Program Committees

- Nataliia Bielova was the PC co-chair (together with Marco Gaboardi) of the ACM workshop "Workshop on Programming Languages and Analysis for Security" (PLAS 2017), which took place in Dallas (USA) on 30 October 2017. http://plas2017.cse.buffalo.edu/

7.1.2.2. Member of the Conference Program Committees

- Ilaria Castellani served in the Program Committees of the conference TTCS'17 and of the workshops PLACES'17 and EXPRESS/SOS'17.
- Tamara Rezk served in the Program Committees of SecDev, Sec@SAC, ESSoS, HotSpot@Etaps.
- Manuel Serrano served the ProWeb'17 program committee.
- Nataliia Bielova served in the Program Committees of CLaw'17, MMM-ACNS'17, ProWeb'17, FCS'17, IEEE SecDev'17, IEEE CSF'17, USENIX Security Poster session, APVP'17.

7.1.2.3. Reviewer

- The team members have been reviewers for the following conferences and workshops: TTCS'17, PLACES'17, EXPRESS/SOS'17, SecDev, Sec@SAC, ESSoS, HotSpot@Etaps, ProWeb'17, CLaw'17, MMM-ACNS'17, ProWeb'17, FCS'17, IEEE SecDev'17, IEEE CSF'17, USENIX Security Poster session, APVP'17.

7.1.3. Journal

7.1.3.1. Member of the Editorial Boards

- Ilaria Castellani is a member of the editorial board of *Technique et Science Informatiques*.
- Ilaria Castellani (together with Mohammad Reza Mousavi) was guest editor for the JLAMP special issue on Trends in Concurrency Theory (Selected invited contributions from the workshops TRENDS 2014 and TRENDS 2015), J. Log. Algebr. Meth. Program., vol. 87, 2017. https://doi.org/10.1016/j.jlamp.2017.01.002.
- Tamara Rezk is a member of the editorial board of *Interstices*.

7.1.3.2. Reviewer - Reviewing Activities

- The team members have been reviewers for the following journals: JLAMP, LMCS, ACM TOIT, ACM Toplas, ACM TOPS.

7.1.4. Invited Talks

- Tamara Rezk was an invited speaker at ETH Zurich in January, at GT-Verif (during the days of the meeting Gdr Sécurité) in May, and at the Journées Scientifiques Inria in June.
- Manuel Serrano gave a keynote talk on Web Programming at the Splash META'17 workshop in September https://2017.splashcon.org/track/meta-2017.
- Nataliia Bielova was an invited speaker at DGCCRF in March, at SySoSec seminar (Rennes) in May, at Higher School of Economics (Moscow) in October, and at DNU and KPI universities (Ukraine) in November.

7.1.5. Leadership within the Scientific Community

- Ilaria Castellani is the chair of the IFIP TC1 WG 1.8 on Concurrency Theory since June 2014.
- Ilaria Castellani is a Management Committee member of the COST Action IC1405 on Reversible Computation (November 2014-November 2018).

7.1.6. Scientific Expertise

- Tamara Rezk was an expert for reviewing projects for the National Center of Science and Technology Agency (Kazakhstan) and the CMU Portugal Program in collaboration with Carnegie Mellon University.

- Nataliia Bielova was an expert for reviewing projects for the Data Transparency Lab (DTL).

7.1.7. Research Administration

- Manuel Serrano is the "Délégué scientifique" (head of research) of the Inria Sophia Antipolis Mediterranée research center. He is member of the Inria Evaluation Committee.
- Nataliia Bielova is a member of “Comité du Suivi Doctoral (CSD)” (Supervision of PhD students) of the Inria Sophia Antipolis Mediterranée research center.
- Ilaria Castellani is a member of Inria’s “Comité Parité et Égalité des Chances”. In the Centre of Inria Sophia Antipolis, she is a member of the “Comité d’Animation et Médiation Scientifique” and of the “Comité Scientifique du Colloquium”.
- Tamara Rezk is a member of the "Commission de Développement Technologique (CDT)" of the Inria Sophia Antipolis Mediterranée research center.

7.2. Teaching - Supervision - Juries

7.2.1. Teaching

Master : Tamara Rezk, Security of Web Applications, 28ETD, niveau M2, University of Nice Sophia Antipolis, France

Master : Tamara Rezk, Preuves en Cryptographie, 28ETD, niveau M2, University of Nice Sophia Antipolis, France

Master : Tamara Rezk, Security of Web Applications, 18ETD, niveau M2, University of Pierre et Marie Curie, France

Master : Nataliia Bielova, Security of Web Applications, 18ETD, niveau M2, University of Pierre et Marie Curie, France

7.2.2. Supervision

PhD in progress: Dolière Francis Somé, Security and Privacy in Web Applications, 1/11/2015, Nataliia Bielova and Tamara Rezk

PhD in progress: Mohamad Ellaz, The DDH Assumption, 1/12/2017, Benjamin Gregoire and Tamara Rezk

PhD in progress: Colin Vidal, *Programmation Web réactive*, University of Nice, 1/07/2015, Manuel Serrano and Gérard Berry.

7.2.3. Juries

- Tamara Rezk was a member of the PhD jury of Zeineb Zhioua, Eurecom.
- Manuel Serrano was a member of the PhD jury of Gabriel Radanne, University Denis Diderot.
- Nataliia Bielova was a member of the PhD jury of Pierre Laperdrix, INSA Rennes.

7.3. Popularization

Tamara Rezk and Nataliia Bielova were contributors to the Security White Report of Inria to provide an overview of the research in the area of IoT security and Web Tracking technologies carried out by Inria teams.

Nataliia Bielova has presented a joint work with Privatics team at the EU Parliament meeting dedicated to the analysis of the new EU ePrivacy Regulation in June 2017. https://www.inria.fr/en/centre/sophia/news/nataliia-bielova-at-the-european-parliament.

Nataliia Bielova gave a general audience talk on Web Tracking technologies and protection mechanisms for Web users at Carnegie-Mellon University (remotely) and SKEMA Business school in October and at Cafe-techno organised by Inria in November.

8. Bibliography

Major publications by the team in recent years

[1] G. BARTHE, T. REZK, A. RUSSO, A. SABELFELD. *Security of Multithreaded Programs by Compilation*, in "ESORICS", 2007, pp. 2-18

[2] N. BIELOVA, T. REZK. *A Taxonomy of Information Flow Monitors*, in "International Conference on Principles of Security and Trust (POST 2016)", Eindhoven, Netherlands, F. PIESSENS, L. VIGANÒ (editors), LNCS - Lecture Notes in Computer Science, Springer, April 2016, vol. 9635, pp. 46–67 [*DOI :* 10.1007/978-3-662-49635-0_3], https://hal.inria.fr/hal-01348188

[3] G. BOUDOL, I. CASTELLANI. *Noninterference for Concurrent Programs and Thread Systems*, in "Theoretical Computer Science", 2002, vol. 281, n^{o} 1, pp. 109-130

[4] G. BOUDOL, Z. LUO, T. REZK, M. SERRANO. *Reasoning about Web Applications: An Operational Semantics for HOP*, in "ACM Transactions on Programming Languages and Systems (TOPLAS)", 2012, vol. 34, n^{o} 2

[5] S. CAPECCHI, I. CASTELLANI, M. DEZANI-CIANCAGLINI. *Typing access control and secure information flow in sessions*, in "Journal of Information and Computation", 2014, vol. 238, pp. 68 - 105 [*DOI :* 10.1016/J.IC.2014.07.005], https://hal.inria.fr/hal-01088782

[6] S. CAPECCHI, I. CASTELLANI, M. DEZANI-CIANCAGLINI. *Information Flow Safety in Multiparty Sessions*, in "Mathematical Structures in Computer Science", 2015, vol. 26, n^{o} 8, 43 p. [*DOI :* 10.1017/S0960129514000619], https://hal.inria.fr/hal-01237236

[7] C. FOURNET, T. REZK. *Cryptographically sound implementations for typed information-flow security*, in "Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2008, San Francisco, California, USA, January 7-12, 2008", 2008, pp. 323-335

[8] M. SERRANO, G. BERRY. *Multitier Programming in Hop - A first step toward programming 21st-century applications*, in "Communications of the ACM", August 2012, vol. 55, n^{o} 8, pp. 53–59 [*DOI :* 10.1145/2240236.2240253], http://cacm.acm.org/magazines/2012/8/153796-multitier-programming-in-hop/abstract

[9] M. SERRANO, E. GALLESIO, F. LOITSCH. *HOP, a language for programming the Web 2.0*, in "Proceedings of the First Dynamic Languages Symposium", Portland, Oregon, USA, October 2006

[10] M. SERRANO. *Bee: an Integrated Development Environment for the Scheme Programming Language*, in "Journal of Functional Programming", May 2000, vol. 10, n^{o} 2, pp. 1–43

Publications of the year

Articles in International Peer-Reviewed Journals

[11] I. CASTELLANI, M. REZA MOUSAVI. *Preface: Special issue on Trends in Concurrency Theory (selected invited contributions from the workshops TRENDS 2014 and 2015)*, in "Journal of Logical and Algebraic Methods in Programming", 2017, vol. 87 [*DOI :* 10.1016/J.JLAMP.2017.01.002], https://hal.inria.fr/hal-01648477

[12] C. ZIELIŃSKI, M. STEFAŃCZYK, T. KORNUTA, M. FIGAT, W. DUDEK, W. SZYNKIEWICZ, W. KASPRZAK, J. FIGAT, M. SZLENK, T. WINIARSKI, K. BANACHOWICZ, T. ZIELIŃSKA, E. G. TSARDOULIAS, A. L. SYMEONIDIS, F. E. PSOMOPOULOS, A. M. KINTSAKIS, P. A. MITKAS, A. THALLAS, S. E. REPPOU, G. T. KARAGIANNIS, K. PANAYIOTOU, V. PRUNET, M. SERRANO, J.-P. MERLET, S. ARAMPATZIS, A. GIOKAS, L. PENTERIDIS, I. TROCHIDIS, D. DANEY, M. ITURBURU. *Variable structure robot control systems: The RAPP approach*, in "Robotics and Autonomous Systems", August 2017, vol. 94, 18 p. [*DOI :* 10.1016/J.ROBOT.2017.05.002], https://hal.inria.fr/hal-01550448

Invited Conferences

[13] T. JENSEN, N. BIELOVA, F. BESSON. *Hybrid information flow analysis against web tracking (invited talk)*, in "The 12th International Conference on Risks and Security of Internet and Systems (CRiSIS 2017)", Dinard, France, September 2017, https://hal.inria.fr/hal-01658896

International Conferences with Proceedings

[14] I. CASTELLANI, M. DEZANI-CIANCAGLINI, P. GIANNINI. *Concurrent Reversible Sessions*, in "CONCUR 2017 - 28th International Conference on Concurrency Theory", Berlin, Germany, CONCUR 2017, Roland Meyer and Uwe Nestmann, September 2017, vol. 85, pp. 1-17 [*DOI :* 10.4230/LIPICS.CONCUR.2017.30], https://hal.inria.fr/hal-01639845

[15] T. REZK, R. CRUZ, B. P. SERPETTE, É. TANTER. *Type Abstraction for Relaxed Noninterference (Artifact) **, in "ECOOP 2017: The 31st European Conference on Object-Oriented Programming", Barcelona, Spain, June 2017, vol. 74, pp. 1 - 7 [*DOI :* 10.4230/DARTS.3.2.9], https://hal.inria.fr/hal-01644835

[16] D. F. SOMÉ, N. BIELOVA, T. REZK. *Control What You Include! Server-Side Protection Against Third Party Web Tracking*, in "International Symposium on Engineering Secure Software and Systems", Bonn, Germany, E. BODDEN, M. PAYER, E. ATHANASOPOULOS (editors), Springer, July 2017, pp. 115-132 [*DOI :* 10.1007/978-3-319-62105-0], https://hal.inria.fr/hal-01649547

[17] D. F. SOMÉ, N. BIELOVA, T. REZK. *On the Content Security Policy Violations due to the Same-Origin Policy*, in "WWW 2017 - 26th International Conference on World Wide Web", Perth, Australia, Proceedings of the 26th International Conference on World Wide Web, ACM, April 2017, pp. 877-886 [*DOI :* 10.1145/3038912.3052634], https://hal.inria.fr/hal-01649526

National Conferences with Proceedings

[18] B. P. SERPETTE, D. JANIN. *Causalité dans les calculs d'événements*, in "JFLA 2017 - Vingt-huitième Journées Francophones des Langages Applicatifs", Gourette, France, January 2017, https://hal.inria.fr/hal-01403369

Scientific Popularization

[19] G. BERRY, J.-P. DELAHAYE. *Jouer ou ne pas jouer au Loto, telle est la stratégie*, in "Interstices", April 2017, https://hal.inria.fr/hal-01533685