The secret language of the internet

What happens when we access a website? In the Software Lab of the Technical University of Darmstadt, researchers are looking for errors in the programs running unnoticed in the background. Their aim is to make the internet more secure and reliable.

By Boris Hänßler

The online service Alexa maintains a list of the most popular websites worldwide. The names at the top, such as Google, YouTube, Facebook, Amazon or Twitter, are familiar to us all. As users of these sites, we rely on the fact that they work perfectly and are secure. But even on these familiar websites, errors in scripts are ever-present, and we are often only aware of them once serious damage has been done.

Michael Pradel from the Software Lab at TU Darmstadt and his team took on the top 100 websites according to Alexa, resolving to search them for any weaknesses. The researchers developed different methods of analysis, which they could use to test a great many websites automatically. „Most people just cannot imagine how many scripts are running in the background all the time as soon as we access an internet site," says Pradel. Scripts are small programs that allow us to interact with website. „And every one of these programs can contain or cause errors. Most of them are harmless, but some are expensive to users."

"We have a lot of ideas on how to extend and refine our methods."

Pradel has his offices on the fourth floor of a building that also houses the Fraunhofer Institute for Secure Information Technology. From his window, he looks out over the city of Darmstadt up to the Mathildenhöhe. His team communicates in English. He did his doctorate in 2012 at ETH Zurich and then was a postdoctoral researcher at the University of Berkeley in California. He moved to Darmstadt in October 2014, to set up the Software Lab.

His focus is on program analysis, and there are three aspects to this as far as he is concerned: reliability, efficiency and data security in web applications. For Pradel, a website is reliable if it acts as the operator intends and does not crash. It is secure if the data of the operator and the site visitors are protected against attack. In addition to this, an ideal site is fast and does not perform any unnecessary calculations.

Pradel's team checks these aspects by means of so-called runtime analyses. "We have developed programs that act like a human being," says Pradel. "But unlike a human user, they systematically trigger all the processes on a site – that is, they simulate all the possible interactions. If a person were to do this, it would sometimes take days or weeks."

Websites have an initial state, so to speak, and that sometimes changes as soon as we move the mouse cursor over a picture. The picture is given a frame, for example, it increases in size, or the mouse cursor becomes a magnifying glass. In these cases, the site moves to a different program state, which in turn allows new interactions. "This produces a huge search space where we can hunt for errors," says Pradel. "The more programs there are behind a site, the more there is to go wrong. Our software cannot explore every script for all known errors, so in each analysis, we concentrate on a specific problem." An example of one such problem is the time it takes the sites to load, which depends to a large extent on how efficiently the scripts are executed. But browsers are not patient. Their limits lie between five and ten seconds. Once this time is exceeded, the browser announces that the site in the window is not available. For the operator, the result can be that a potential customer is annoyed, and takes their business to the competition.

The programmers of websites struggle to detect this type of error early on. They sometimes only occur under certain circumstances, such as when the user takes a number of steps in a fixed order. So the errors are overlooked during a routine check of the site. The Software Lab analysis software keeps calling up the program codes in a different order, so that nothing escapes the test.

Errors in connection with JavaScript are among the most common security risks on the internet. "JavaScript is an accident waiting to happen", says

An invitation to hackers

Cases of problematic weaknesses that can be exploited by hackers keep being made public:
Two years ago, when Twitter introduced a new layout, the programmers overlooked the fact that users could send links with JavaScript instructions. As soon as the Twitter users moved their mouse cursors over these links, the problematic tweets were forwarded from the individual account to all its followers – so that vast numbers of the links were circulated within minutes, before Twitter found the loophole.

In 2014, the auction platform eBay also fell victim to this so-called cross-site-scripting (XSS): A user had an iPhone for sale. But when those interested in buying clicked on the site link, they were diverted to an external, fake website, where they were asked to enter their eBay login data again. This data was intercepted by the hackers. At the start of 2016, eBay again had to admit to having discovered such a loophole, and closed it.

Information

Software Lab Research Group in the Department of Computer Science
Dr Michael Pradel
Phone: ++49 (0)6151/86 95 06
E-mail:
michael@binaervarianz.de
www.sola.tu-darmstadt.de

Photo: Katrin Binner

Dr Michael Pradel, Head of the Software Lab research group.

Pradel. “The language was introduced more than 20 years ago. At the time, a Netscape engineer had ten days to develop a language for interaction with the browser. As the language is backward compatible, all its shortcomings from those times are still present.” JavaScript is popular because the language is easy to learn and websites do not crash if there are errors in the script – the browsers simply ignore the errors. Other programming languages need compilers, which stop the program immediately in the event of an error and indicate the relevant point in the code. They force programmers to work more carefully.

Not all shortcomings are caused by the carelessness of developers. Sometimes scripts from other sources such as advertisements, or embedded videos are loaded onto the websites. The programmers cannot see these scripts in advance. Other errors that often do not come to light during routine checks are caused by “data races”. Two parts of a JavaScript program get in each other’s way, because they can be executed in random order and store different data in the same memory. The subsequent calculation can change, depending on which part of the program is first. So it can happen that a book in a shop suddenly costs 2,000 Euros, instead of 15 Euros.

These are just a few examples. “We are nowhere near finishing all the analytical work”, says Pradel. “We have a lot of ideas on how to extend and refine our methods.” Pradel’s team has already written to some of the top-100 website operators to point out errors. A few companies have remained silent, while others have been quick to correct the weaknesses. Although we are unaware of all that goes on, we thank the Software Lab for saving us from bother on the internet.

The author is a technology journalist.

Cybersecurity profile area

The Software Lab is a research group in the Department of Computer Science at TU Darmstadt led by Dr Michael Pradel. Its research scientists explore tools and techniques that help to develop reliable, efficient, and secure software. The Software Lab is part of the cybersecurity profile area at TU Darmstadt. Teams from eight of the thirteen university departments work here on challenges related to cybersecurity and privacy protection.