Die heimliche Netzsprache

Was passiert, wenn wir eine Webseite aufrufen? Am Software Lab der Technischen Universität Darmstadt suchen Forscher nach Fehlern in den Programmen, die dabei unbemerkt im Hintergrund ablaufen. Das Ziel ist ein sicheres und zuverlässiges Internet.

Von Boris Hänßler

Der Online-Dienst Alexa pflegt eine Liste mit den weltweit populärsten Webseiten. An der Spitze stehen Namen, die wir alle kennen, zum Beispiel Google, YouTube, Facebook, Amazon oder Twitter. Als Nutzer dieser Angebote verlassen wir uns darauf, dass sie einwandfrei funktionieren und sicher sind. Aber fehlerhafte Skripte sind selbst auf diesen bekannten Webseiten allgegenwärtig, nur oft bekommen wir das erst mit, wenn ein ernster Schaden entstanden ist.

Michael Pradel vom Software Lab der TU Darmstadt hat sich mit seinem Team die Top 100-Webseiten laut Alexa vorgenommen und dort nach Schwachstellen gefahndet. Die Forscher entwickelten verschiedene Analyseverfahren, mit der sie eine große Zahl an Webangeboten automatisch testen können. „Für die meisten Menschen ist es unvorstellbar, wie viele Skripte jedes Mal im Hintergrund ablaufen, sobald wir eine Internetseite aufrufen“, sagt Pradel. Skripte sind kleine Programme, die es uns ermöglichen, mit der Webseite zu interagieren. „Und jedes dieser Programme kann Fehler enthalten oder verursachen. Die meisten sind harmlos, aber einige gehen auf Kosten der Nutzer.“

„Wir haben viele Ideen, wie wir unsere Methoden ausweiten und verfeinern können.“

Pradel sitzt im vierten Obergeschoss eines Gebäudes, in dem auch das Fraunhofer-Institut für Sichere Informationstechnologie untergebracht ist. Von seinem Fenster aus kann er über die Stadt Darmstadt bis zur Mathildenhöhe blicken. Mit seinem Team redet er auf Englisch. Er hat 2012 an der ETH Zürich promoviert und anschließend als Postdoc an der Universität Berkeley in Kalifornien geforscht. Im Oktober 2014 kam er nach Darmstadt, um das Software Lab aufzubauen.

Sein Schwerpunkt ist die Programmanalyse, und für ihn umfasst sie drei Aspekte: Zuverlässigkeit, Effizienz und Datensicherheit bei Webapplikationen. Für Pradel ist eine Webseite zuverlässig, wenn sie sich so verhält, wie vom Betreiber intendiert, und nicht abstürzt. Sie ist sicher, wenn die Daten des Betreibers und der Besucher gegen Angriffe geschützt sind. Außerdem ist eine optimale Seite schnell und führt keine unnötigen Rechenoperationen durch.

Pradels Team prüft diese Aspekte anhand sogenannter Laufzeitanalysen. „Die von uns entwickelten Programme verhalten sich wie ein Mensch“, sagt Pradel. „Aber im Gegensatz zu einem menschlichen Nutzer sollen sie systematisch sämtliche Abläufe auf einer Seite auslösen – also alle möglichen Interaktionen simulieren. Würde das ein Mensch machen, bräuchte er mitunter Tage oder Wochen.“

Webseiten haben sozusagen einen Startzustand und der verändert sich mitunter schon, wenn wir den Mauszeiger nur über ein Bild bewegen. Das Bild erhält zum Beispiel einen Rahmen, es wird vergrößert oder als Mauszeiger erscheint eine Lupe. Die Seite geht in diesen Fällen auf der Programmebene in einen anderen Zustand über, der wiederum neue Interaktionen zulässt. „Dabei entsteht ein riesiger Suchraum, den wir nach Fehlern durchforsten können“, sagt Pradel. „Je mehr Programme hinter der Seite liegen, desto mehr kann schief gehen. Unsere Software kann nicht jedes Skript auf alle bekannten Fehler hin untersuchen, daher konzentrieren wir uns bei jeder Analyse auf ein bestimmtes Problem.“ Ein Beispiel für so ein Problem ist die Ladezeit von Seiten, die stark davon abhängt, wie effizient die Skripte ausgeführt werden. Aber Browser sind nicht geduldig. Sie haben Grenzwerte zwischen fünf und zehn Sekunden. Ist diese Zeit überschritten, verkündet der Browser, dass die Seite im Fenster nicht zur Verfügung stehe. Für den Betreiber kann dies zur Folge haben, dass ein potentieller Kunde verärgert zur Konkurrenz wechselt.

Die Programmierer von Webseiten tun sich schwer damit, solche Fehler frühzeitig zu erkennen. Sie treten

Einladungen an Hacker

Immer wieder werden Fälle von problematischen Schwachstellen publik, die Hacker ausnutzen konnten: Als vor einigen Jahren Twitter ein neues Layout vorstellte, übersahen die Programmierer, dass die Nutzer Links mit JavaScript-Anweisungen versenden konnten. Sobald Twitter-Nutzer den Mauszeiger über einen solchen Link bewegten, wurden die problematischen Tweets vom eigenen Account aus an alle Follower weiter geleitet – so dass sich die Links innerhalb von Minuten massenhaft verbreiten konnten, ehe Twitter die Lücke fand.

Auch die Versteigerungsplattform Ebay wurde 2014 Opfer eines solchen sogenannten Cross-Site-Scriptings (XSS): Ein User hatte ein iPhone zum Verkauf eingestellt. Aber als interessierte Käufer auf den Angebotslink klickten, wurden sie auf eine externe, gefälschte Seite umgeleitet und dort aufgefordert, ihre Ebay-Login-Daten neu einzugeben. Diese Daten wurden von den Hackern abgefangen. Anfang 2016 musste Ebay erneut eingestehen, eine solche Lücke entdeckt und geschlossen zu haben.

Informationen

Forschungsgruppe Software Lab im Fachbereich Informatik
Dr. Michael Pradel
Telefon: 06151/86 95 06
E-Mail:
michael@binaervarianz.de
www.sola.tu-darmstadt.de

Abbildung: Katrin Binner

Dr. Michael Pradel, Leiter der Nachwuchsgruppe Software Lab.

mitunter nur unter bestimmten Voraussetzungen auf, zum Beispiel wenn der User mehrere Schritte in einer festen Reihenfolge ausführt. Beim Routine-Check der Seite werden die Fehler deshalb übersehen. Die Analyse-Software vom Software Lab ruft die Programmcodes immer wieder in unterschiedlicher Reihenfolge ab, damit der Prüfung nichts entgeht.

Fehler im Zusammenhang mit JavaScript zählen zu den häufigsten Sicherheitsrisiken im Internet. „JavaScript ist eine Art Unfall", sagt Pradel. „Die Sprache wurde vor mehr als 20 Jahren eingeführt. Ein Ingenieur von Netscape hatte damals zehn Tage Zeit, eine Sprache für Interaktionen mit dem Browser zu entwickeln. Da die Sprache abwärtskompatibel ist, sind alle Mängel von damals noch vorhanden." JavaScript ist beliebt, weil die Sprache einfach zu lernen ist und weil Webseiten bei fehlerhaftem Skript nicht abstürzen – die Fehler werden von den Browsern ignoriert. Andere Programmiersprachen benötigen Compiler, die bei Fehlern das Programm sofort anhalten und auf die entsprechende Stelle im Code hinweisen. Sie zwingen Programmierer, sorgfältiger zu arbeiten.

Nicht alle Mängel haben ihre Ursache in der Unachtsamkeit der Entwickler. Manchmal werden auf Webseiten Skripte geladen, die von anderen Quellen kommen, zum Beispiel Werbeanzeigen oder eingebettete Videos. Die Programmierer können solche Skripte nicht vorher einsehen. Andere Fehler, die bei Routinechecks häufig nicht auffallen, entstehen durch „Data Races". Dabei kommen sich zwei Teile eines JavaScript-Programmes in die Quere, weil sie in zufälliger Reihenfolge ausgeführt werden können und dabei unterschiedliche Daten im selben Speicher ablegen. Je nachdem, welcher Programmteil zuerst dran ist, ändert sich die anschließende Berechnung. So kann es vorkommen, dass in einem Shop ein Buch statt 15 Euro plötzlich 2.000 Euro kosten soll.

Das sind nur einige Beispiele. „Die Analysearbeit wird uns so schnell nicht ausgehen", sagt Pradel. „Wir haben viele Ideen, wie wir unsere Methoden ausweiten und verfeinern können." So manchen Betreiber der Top 100-Webseiten hat Pradels Team übrigens angeschrieben und auf Fehler hingewiesen. Einige Firmen schwiegen, andere korrigierten die Schwachstellen schnell. Obwohl wir von alldem nichts mitbekommen, verdanken wir dem Software Lab, dass uns einiger Ärger im Netz erspart bleibt.

Der Autor ist Technikjournalist.

Profilbereich Cybersicherheit
Das Software Lab ist eine Forschungsgruppe des Fachbereiches Informatik der TU Darmstadt, geleitet von Dr. Michael Pradel. Die Wissenschaftlerinnen und Wissenschaftler erforschen Werkzeuge und Methoden, die helfen, zuverlässige, effiziente und sichere Software zu entwickeln. Das Software Lab ist Teil des Profilbereichs Cybersecurity der TU Darmstadt. Hier arbeiten Teams aus acht der 13 Fachbereiche der Universität an zentralen Themen der Cybersicherheit und des Schutzes der Privatheit.