New tool aims to ensure software security policies reflect user needs

October 30 2012

Researchers from North Carolina State University and IBM Research have developed a new natural language processing tool that businesses or other customers can use to ensure that software developers have a clear idea of the security policies to be incorporated into new software products.

Specifically, the research focuses on access control policies (ACPs), which are the security requirements that [software developers](#) need to bear in mind when developing new software. For example, an ACP for a university grading program needs to allow professors to give grades to students, but should not allow students to change the grades.

"These ACPs are important, but are often buried amidst a lengthy list of other requirements that customers give to developers," says Dr. Tao Xie, an associate professor of [computer science](#) at NC State and co-author of a paper on the research. These requirements are written in "natural language," which is the conversational language that people use when talking or corresponding via the written word.

Incomplete or inaccurate ACP requirements can crop up, for example, if the customer writing the ACP requirements makes a mistake or doesn't have enough technical know-how to accurately describe a program's security needs.

A second problem is that programmers may misinterpret some ACP requirements, or overlook them entirely.

In collaboration with IBM Research, Xie's research team has developed a solution that uses a [natural language processing](#) program to extract the ACP requirements from a customer's overall list of requirements and translate it into machine-readable language that computers can understand and enforce.

After the ACPs are extracted, they can be run through Access Control Policy Tool (ACPT) – also developed in Xie's research team in collaboration with the National Institute of Standards and Technology (NIST) – which verifies and tests the ACPs and determines whether the ACP requirements are adequate to meet the security needs of the program.

Once the ACP requirements have been translated into machine-readable language, they can also be incorporated into a policy-enforcement "engine" in the final software product – which ensures that ACPs cannot be overlooked by programmers.

"In general, developing a program that understands natural language text is very challenging," Xie says. "However, ACP requirements in software documents usually follow a certain style, using terms such as 'cannot be edited' or 'does not have the ability to edit.' Because ACPs tend to use such a limited number of phrases, it is much easier to develop a program that effectively translates natural language texts in this context."

  More information: [people.engr.ncsu.edu/txie/publ …](#) [ations/fse12-nlp.pdf](#)

Provided by North Carolina State University