Trust in global computing

July 12 2006

Access to distributed mobile resources by software agents of all types promises much for global computing. But it suffers from the same security and trust problems as the internet itself. Now new tools and protocols could lay the foundations for new and more secure high-level global computing.

But what does 'global computing' actually mean? The term refers to abstractions such as overlay computers, which involves access to distributed mobile resources by software agents that are not tied to a specific geographical or logical network location.

The key challenge for systems designers and programmers in such scenarios is that the software agents have to operate within environments about which they have little information, and where other agents are not necessarily trustworthy.

“Your piece of software, alone and out there in the wild, doesn’t know who to trust and who not!” says Vladimiro Sassone of the University of Southampton, coordinator of the MYTHS project financed under the European Commission’s Future and Emerging Technologies (FET) initiative of the IST programme. “That is why closed networks exist. In a global computing environment you do not have the reassurance of a closed network – you are dealing with agents that you cannot trust. That is why security issues are paramount.”

However, if the global market for internet services and applications is to progress, systems designers need to develop adequate security guarantees for agent-based transactions. Specific domains have to be able to limit access to selected agents only. These agents also need to protect themselves and their data from attacks while traversing potentially hostile environments, or executing remotely outside the control of their originating locations. In other words, you need to give your car keys with a guarantee that the car will arrive (eventually) on your doorstep.

MYTHS, which completed in March 2005, sought specifically to develop ‘type’-based theories of security for mobile and distributed systems, as a possible route to solving such problems. “Types are fundamentally important because they express a property of a particular program or piece of code which is unvarying,” says Sassone. “Their other important property is that they can be checked by inspecting the code rather than running the program, which may be too big and complex to run easily.”

Starting with the principle that strong typing underpins truly secure computing applications, the MYTHS team focused on the foundations of programming languages and the paradigms that allow static detection of security violations. The team aimed to develop type-based methods and tools that would allow computing systems and applications to be formally analysed for security weaknesses.

Results in three key areas

Their results can be divided into three main areas. In resource access-control, in other words how to control access by software agents to specific resources, “We developed complex type systems to control access to certain resources – the type determines that a specific piece of code would never be able to migrate to certain areas of your network,” says Sassone. “For example the code could migrate to online shop one because you trust that outlet, but not to online shop two.”

In crypto-protocol analysis, cryptographic protocols are delicate and vulnerable to attack. Many protocols may actually reveal the content of the code by disclosing the behaviour of the system. Such protocols have in the past not been sufficiently well-designed to resist the more sophisticated forms of attack.

“We designed a tool called PEAR,” says Sassone, “which analyses protocol specifications by assigning types to various messages. The tool enables systems programmers to analyse how secure a protocol is, and to see if it will leak information when under attack.”

In the area of data manipulation, the project team developed a brand-new programming language for the manipulation of XML documents, facilitating the examination and analysis of XML data. The language, CDuce, is an innovative XML-oriented functional language which is type-safe, efficient and offers new methods of working with XML documents. A compiler is also available under an open-source licence.

Sassone emphasises that the work within MYTHS dealt with the foundations, with computational theory, but that it can nevertheless underpin real tools. “Types can be implanted in programming languages, to deliver code that can work out there in the real world.”

The PEAR tool for analysing cryptographic protocols has been further improved since the close of the project, and has been presented at several EU fora. Another key project result, the new CDuce XML programming language, has generated a great deal of interest. So much so that the project researcher specialising in this area is now working full-time on its further development.

Source: IST Results