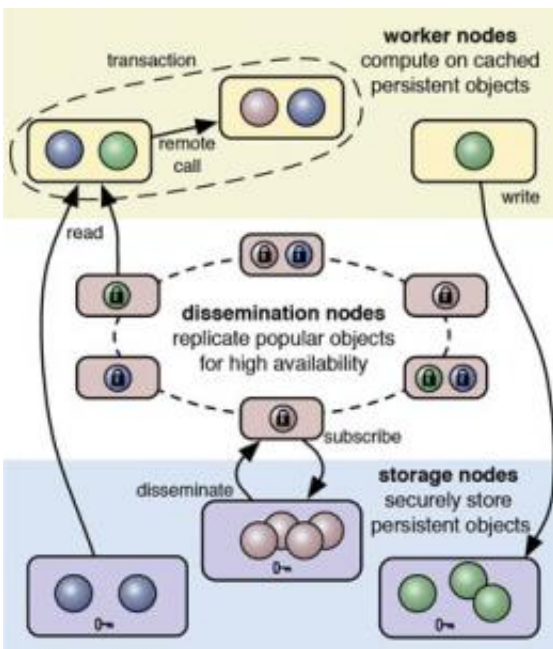# 'Fabric' would tighten the weave of online security

October 1 2010, By Bill Steele

Nodes (locations on a computer network) in Fabric pass around objects that contain data and program code, but the objects have built-in rules about what each node can do with them. The Fabric language requires programmers to include these rules and saves them the work of writing code to enforce them.

(PhysOrg.com) -- As we become increasingly dependent on computers to manage our lives and businesses, our money and privacy become less and less secure. But now, Cornell researchers offer a way to build security into computer systems from the start, by incorporating security in the language used to write the programs.

Until now, computer security has been reactive, said Fred Schneider, the Samuel B. Eckert Professor of Computer Science. When hackers discover a way in, we patch it. "Our defenses improve only after they have been successfully penetrated," he explained.

"When problems arise, we patch software like putting on duct tape," added collaborator Andrew Myers, professor of computer science. "By now we have layers of duct tape, and the system is a mess. ... Our computer systems are this tottering stack of obsolete [layers of software] ... and security vulnerabilities are nearly inevitable."

Myers and Schneider are developing a new computer platform, dubbed "Fabric," that replaces multiple existing layers with a single, simpler programming interface that makes security reasoning explicit and direct, Myers said.

Fabric is designed to create secure systems for distributed computing, where many interconnected nodes -- not all of them necessarily trustworthy -- are involved, as in systems that move money around or maintain medical records. When you connect to Amazon, for example, it talks to your credit card company and the vendor of the product, passes your demographics to some advertisers and more. In a medical records system, data is shared between hospitals, doctors and other practitioners, laboratories, medical billing agencies and insurers.

Fabric's programming language, an extension of the widely used Java language, builds in security as the program is written. Everything in Fabric is an "object" labeled with a set of policies on how and by whom data can be accessed and what operations can be performed on it. Even blocks of program code have built-in policies about when and where they can be run.

While your medical record, for example, could be seen entirely by your

doctor, your physical therapist might be able to see only the doctor's prescription for your therapy, and your insurance company could see only the charges.

The compiler that turns the programmer's code into an executable program enforces the security policies and will not allow the programmer write insecure code, Myers said. Most of this, he added, is transparent to the programmer, who can simply set the policies and not have to write detailed code to enforce them. "I think we can make life simpler and improve performance," he said.

Fabric is still a prototype, being tested on a database of Cornell computer science students. With $1.1 million from the National Science Foundation and $1.3 million from the Office of Naval Research, Schneider and Myers plan to scale it up for very large distributed systems, provide for more complex security restrictions on objects and enable "mobile code" -- programs that can reside on one node of a network and be run on another with assurance that they are safe and do what they claim to do. And perhaps most important (and perhaps hardest), they hope to provide formal mathematical proof that a system is really secure.

Will the computer establishment be willing to adopt this new way of managing complex systems? "How did we get people to use the Web?" Myers countered. "It's a paradigm shift. By making security policies part of the process of building software, we can make it much easier to build secure systems. That will drive adoption."

The name "Fabric," he noted, is meant to be reminiscent of "the Web," but "Fabric is more useful and more tightly connected than webs."

Provided by Cornell University