PROTECTION AND SECURITY : IMPLICATIONS

What sort of facilities must a system provide as a basis for a satisfactory level of protection and security ? That's really two questions – while protection and security are akin in their end of keeping people's computing safe, they differ in the means they employ. When providing protection, our concern is to help people get things right, and we can expect them to cooperate – or, at worst, to ignore the devices we provide. In contrast, a security mechanism is specifically designed to prevent people doing things we think they shouldn't, and it must be able to withstand their deliberate attack.

What that means is, roughly, that we can provide some sort of protection for any computer system; but there is no hope of imposing security on a machine which doesn't incorporate some provision for security in its hardware. At the very least, people without special privileges must not be able to construct programmes which will in any way reach outside their own part of the system. Security can't be created from nothing; the hardware must provide some sort of elementary security machinery if any form of security is to work.

FOR PROTECTION :

Information.

Many protection mechanisms depend on the system's having information about what it's doing. As most things in the system are files of one sort or another, file attributes (in principle, any information about files) are an important part of the system's information. Particular examples are access rights (who can do what with a file) and file type (what sort of a file it is – code file, character file, dot-matrix picture).

Archiving and Backup.

Discretionary archiving and primitive backup facilities can be added to a conventional file system without any special additional provisions; but both automatic archiving and system backup require specific information from the file system.

Hardware.

Memory protection is important for safe computing, and can only be provided with any sort of efficiency by hardware. The same is true for array indexing range checks, and arithmetic overflow checks. While it's just about possible, if you're desperate, to check for indexing and overflow in software, the only really satisfactory way to implement checks which should always be made is to build them into the hardware, so that they can operate in parallel with the normal processing.

Restrictions.

We're not sure whether or not you can count an absence as a facility, but a significant contribution to system safety is to have no assembler (though you need something much stronger than protection techniques if you want to prevent determined people from writing their own assemblers).

FOR SECURITY :

Identification.

An important prerequisite for security is to know who is doing what at all times; so anybody using the system must be identified. Really secure systems might use special hardware to help in identification.

A secure file system.

The file system is where the operating system keeps all the information it needs to implement security : obviously, this information must be known to be safe from unauthorised access.

Hardware.

Even the most common protection mechanisms – supervisor calls – depend on hardware features for their effective implementation. Capability schemes of the tagged architecture type also need hardware assistance to prevent unauthorised computing with the capability variables.

Hardware can also help you to prevent people from writing assemblers – but it's expensive.
