

# IY5512: Coursework 2: Worked solutions

---

1. *The following quote is taken (in slightly edited form) from the Wikipedia article on 'Principle of least privilege' (as of 15/10/11):*

*The principle of least privilege is widely recognised as an important design consideration in enhancing the protection of data and functionality against faults (fault tolerance) and malicious behaviour (computer security).*

*Describe a practical example of a use of this principle.*

There are many examples on the web – just search for 'least privilege'. One is reproduced below. The following article is an edited version of text taken from:

<http://www.brighthub.com/computing/smb-security/articles/39545.aspx>

In Information Security theory the principle of least privilege states that users and processes should only have access to the minimum amount of information and functions required to do their job. 'Vertical' escalation of privileges can arise when an ordinary user assumes the privileges of an administrator, service, or kernel, and such an escalation can clearly be a security threat. 'Horizontal' escalation is also a concern, as user *A* might obtain the privileges of user *B* and thereby gain access to user *B*'s valuable data. De-escalation can also be a concern, where an administrative user that is normally segregated (or separated) from data accessible to less privileged users obtains the privilege(s) needed to access this data.

Some applications act with the privilege of the user running them. This is usually as it should be. However, other applications, such as services (in Windows) or daemons (in Unix systems) may have higher or different privilege levels. Some applications may need to temporarily assume a higher privilege level, then 'drop privileges' to that of an ordinary user. Others may interact with a high-privilege level service or part of the OS or kernel.

While an application has a higher privilege level, a bug or other vulnerability in the application code may allow a user to act as the application and use its higher privileges to access information that is normally not available to that user. Often application or system crashes have been used in this way. Buffer overruns, for example, can allow code to be injected by a hacker, and this custom code then executes with the higher privileges and can thus accomplish tasks that the user otherwise could not. In these cases the vulnerability is often discovered by hackers and a proof of concept exploit is created and shown to the developer(s). Ideally a patch will then be forthcoming to remove the vulnerability.

Carefully architecting security groups and restrictions, segregating users, and isolating tiers or layers of access are design choices that can assist in preventing privilege escalation. Often the root of the problem is in bugs or errors in operating systems and application coding. What can we do about these problems? The simple answer is to test thoroughly, keep patches current, and when problems are discovered deal with them promptly.

2. *Security evaluations can be performed by government agencies (as in the US) or by commercial evaluation facilities (as in Europe). Discuss the relative advantages and disadvantages of the two approaches.*

Commercial evaluations tend to be very expensive. As a result relatively few products are evaluated, and the cost of the evaluation will be reflected in the end-user price. Moreover there is a danger that inconsistencies may arise between different commercial evaluation bodies, particularly given that they will be subject to commercial pressures. Hence in a commercial environment particular care must be taken in framing the criteria and also in checking the consistency of the work of the bodies.

Evaluations by government agencies have historically been free. However, as a result large backlogs of products awaiting evaluation have built up. Whilst consistency checking may be less of a problem, there is still a danger of 'interpretation drift' (sometimes called 'criteria creep') over time. Historically, when evaluation was primarily intended for products to be sold to the government, a government run and subsidised service made a lot of sense. However, if, as is the intention of the Common Criteria, evaluation is aimed at a larger market, then the adoption of a commercial evaluation model is probably inevitable.

3. *Discuss the difference between products and systems, as defined in ITSEC. Describe how evaluation of both products and systems is made possible through the notion of a Target of Evaluation (ToE), and, in doing so, describe what might be contained in a ToE.*

A product is something which can be bought 'off the shelf'. A system may involve a number of products, which may be tailored to a particular environment.

To enable the evaluation process to operate, both must have a specified 'Target of Evaluation'. The security target within the ToE would typically contain the following:

- security objectives,
- statements about the system environment,
- assumptions about the ToE environment,
- security functions,
- rationale for security functions,
- required security mechanisms,
- required evaluation level,
- claimed rating of the minimum strength of the mechanisms.