Administrative Interfaces

Objective

To ensure that

* administrator level functions are appropriately segregated from user activity
* Users cannot access or utilize administrator functionality
* Provide necessary audit and traceability of administrative functionality

Environments Affected

All.

Relevant COBIT Topics

PO4

* 4.08 Data and System ownership – requires separate operational and security administration
* 4.10 Segregation of duties

Best practices

Administrative interfaces is one of the few controls within the Guide which is legally mandated – Sarbanes Oxley requires administrative functions to be segregated from normal functionality as it is a key fraud control. For organizations that have no need to comply with US law, ISO 17799 also strongly suggests that there is segregation of duties. It is obviously up to the designers to take into account the risk of not complying with SOX or ISO 17799.

* When designing applications, map out administrative functionality and ensure that appropriate access controls and auditing are in place.
* Consider processes – sometimes all that is required is to understand how users may be prevented from using a feature by simple lack of access
* Help desk access is always a middle ground – they need access to assist customers, but they are not administrators.
* Carefully design help desk / moderator / customer support functionality around limited administration capability and segregated application or access if possible

This is not to say that administrators logging on as users to the primary application are not allowed, but when they do, they should be normal users. An example is a system administrator of a major e-commerce site who also buys or sells using the site.

Administrators are not users

Description

Administrators must be segregated from normal users.

How to identify if you are vulnerable

* Log on to the application as an administrator.
* Can the administrator perform normal transactions or see the normal application?
* Can users perform administrative tasks or actions if they know the URL of the administration action?
* Does the administrative interface use the same database or middleware access (for example, database accounts or trusted internal paths?)
* In a high value system, can users access the system containing the administrative interface?

If yes to any question, the system is potentially vulnerable.

How to protect yourself

All systems should code separate applications for administrator and user access. High value systems should separate these systems to separate hosts, which may not be accessible to the wider Internet without access to management networks, such as via the use of a strongly authenticated VPN or from trusted network operations center

Authentication for high value systems

Description

Administrative interfaces by their nature are dangerous to the health of the overall system. Administrative features may include direct SQL queries, loading or backing up the database, directly querying the state of a trusted third party’s system.

How to identify if you are vulnerable

If a high value system does not use strong authentication and encrypted channels to log on to the interface, the system may be vulnerable from eavesdropping, man in the middle, and replay attacks.

How to protect yourself

For high value systems:

* Use strong authentication to log on, and re-authenticate major or dangerous transactions to prevent administrative phishing and session riding attacks.
* Use encryption (such as SSL encrypted web pages) to protect the confidentiality and integrity of the session.

Further Reading

TODO