Hardening (computing)

In computer security, **hardening** is usually the process of securing a system by reducing its surface of vulnerability, which is larger when a system performs more functions; in principle a single-function system is more secure than a multipurpose one. Reducing available ways of attack typically includes changing default passwords, the removal of unnecessary software, unnecessary usernames or logins, and the disabling or removal of unnecessary services.

There are various methods of hardening Unix and Linux systems. This may involve, among other measures, applying a patch to the kernel such as Exec Shield or PaX; closing open network ports; and setting up intrusion-detection systems, firewalls and intrusion-prevention systems. There are also hardening scripts and tools like Lynis, Bastille Linux, JASS for Solaris systems and Apache/PHP Hardener that can, for example, deactivate unneeded features in configuration files or perform various other protective measures.

Binary hardening

Binary hardening is a security technique in which binary files are analyzed and modified to protect against common exploits. Binary hardening is independent of compilers and involves the entire toolchain. For example, one binary hardening technique is to detect potential buffer overflows and to substitute the existing code with safer code. The advantage of manipulating

binaries is that vulnerabilities in legacy code can be fixed automatically without the need for source code, which may be unavailable or obfuscated. Secondly, the same techniques can be applied to binaries from multiple compilers, some of which may be less secure than others.

Binary hardening often involves the non-deterministic modification of control flow and instruction addresses so as to prevent attackers from successfully reusing program code to perform exploits. Common hardening techniques are:

- Buffer overflow protection
- Stack overwriting protection
- Position independent executables and address space layout randomization
- Binary stirring (randomizing the address of basic blocks)
- Pointer masking (protection against code injection)
- Control flow randomization (to protect against control flow diversion)

See also

- Computer security
- Computer network security
- Network security policy
- Security-focused operating system
- Security-Enhanced Linux

References

External links

- "Hardening Your Computing Assets" (http://www.globalsecurity.org/military/library/report/1997/harden.pdf) (PDF). at globalsecurity.org