Buffer overflow and rootkits

Technology, Information Technology

Buffer Overflow & Rootkits al Affiliation Buffer Overflow & Rootkits Computer systems are vulnerable to attacks by a variety of on the internet. Most people’s computers become the victims of these attacks because of the lack of enquired protection. The reason is the lack of awareness of the existence of protective programs that might avert these attacks. Other individuals are just ignorant that such attacks even take place in the first place and, therefore, see no reason to install protective software (Ben-shalom, Naystut & Muttik, 2015). Perhaps, this is due to the nature of some attacks to integrate themselves well into the computer’s operating system. Another reason is the failure of individuals to update their existing protections that render them non-functional.

Someone can design a program that can detect a rootkit. A rootkit describes stealthy software that makes it hard to detect the existence of certain programs in a computer. Therefore, to detect one, someone needs to use an alternative and trusted operating system. Designing programs constituting such operating systems would make it possible to detect rootkits. Other ways include signature scanning, difference scanning, behavioural-based methods and memory dump analysis.

The behavioural-based approach depends on the fact that rootkits behave in a way different from other programmes (Pleeger, 2012). In signature scanning, an antivirus will detect any stealthy measures that a rootkit might adopt to unload itself or prevent its detection. The difference-based scanning method compares trusted original data from the computer with defective data returning from the API (Application programming interface), a programme building tool. Memory dumping involves dumping of virtual

memory, which can then be forensically analysed to capture an active rootkit with a tool called a debugger. It prevents the rootkit from taking any measures to hide itself. However, the overall detection of a rootkit depends on its sophistication.

References

Ben-shalom, O., Naystut, A. & Muttik, I. (2015). U. S. Patent No. 20, 150, 007, 316. Washington, DC: U. S. Patent and Trademark Office

Pleeger, C. (2012). Analyzing Computer Security: A Threat / Vulnerability / Countermeasure Approach. New York: Prentice Hall-Pearson Publication.