Portably solving the access(2)/open(2) race

Stephen Checkoway
Johns Hopkins University

ABSTRACT

The access(2)/open(2) file-system race is the canonical example of a time-of-check-to-time-of-use (TOCTTOU) error in which a setuid binary checks that a user has permission to open a file prior to opening it. By changing the state of the file-system between the calls to access(2) and open(2), an attacker can cause the program to open a file to which the user does not have access. This race has been the focus of several papers alternately trying to defend against the race and attacking the defenses [3, 1, 2]. In this paper, we give a simple solution that avoids the race condition in all POSIX.1-conformant operating systems such as Mac OS X 10.6.8, Linux 2.6.35, FreeBSD 8.2, NetBSD 5.1, OpenBSD 4.9, Dragonfly BSD 2.10.1, and Solaris 10. In other words, most modern, UNIX-like operating systems.

Dean and Hu explicitly reject a solution based on temporarily changing user ids writing, "a solution depending on user id juggling can be made to work, but is generally not portable" [3]. This may have been true in 2004, but is no longer the case today as all modern UNIX-like operating systems correctly implement the seteuid(2)/setegid(2) system calls—note the 'e' for "effective." Rather than testing if a file can be opened and then opening it, a secure setuid program should use seteuid(2)/setegid(2) and simply open(2) the file. If the open(2) call fails with errno set to EPERM, then the user/group did not have permission to open the file. If the call succeeds, then the user/group had permission.

BODY

access(2)/open(2) file-system races can be prevented by omitting access(2) and using seteuid(2)/setegid(2) before open(2) on a modern OS.

REFERENCES

[1] N. Borisov, R. Johnson, N. Sastry, and D. Wagner. Fixing races for fun and profit: How to abuse atime. In P. McDaniel, editor, *Proceedings of USENIX Security 2005*. USENIX, Aug. 2005.

[2] X. Cai, Y. Gui, and R. Johnson. Exploiting unix file-system races via algorithmic complexity attacks. In A. Myers and D. Evans, editors, *Proceedings of IEEE Symposium on Security and Privacy ("Oakland") 2009*. IEEE Computer Society, May 2009.

[3] D. Dean and A. J. Hu. Fixing races for fun and profit: How to use *access(2)*. In M. Blaze, editor, *Proceedings of USENIX Security 2004*. USENIX, Aug. 2004.

Volume 1 of Tiny Transactions on Computer Science