CERIAS

Refactoring Secure Programming Classes

Pascal Meunier[1], Jared Robinson[2] and Matt Rose[1]

1: Purdue University CERIAS; 2: Symantec Corporation

Abstract

Undergraduate curricula most often focuses on achieving functional objectives, and assignments are graded accordingly. Security objectives are typically not emphasized, so most software vulnerabilities are repeated, well-understood mistakes. Secure programming classes have been taught at Purdue since the Fall 2002 in order to address this problem. However, the scope of the classes was limited to that of an associated "sister" class (e.g., operating systems), and biased towards UNIX. We expanded the scope of the previously developed material to include Windows issues and created exercises and labs that are independent of any "sister" class. We also added coverage of trust, threat, risk issues as well as software engineering considerations. In the process we uncovered Windows threats that have been neglected in the litterature. The material is freely available at http://www.cerias.purdue.edu/secprog/ as sets of slides and is divided into three modules, which can be taught to different audiences.

Organization and Audience

The material is separated into three modules. The first module presents security concepts such as trust and assurance, and is aimed at anyone involved in the development process, including managers.

Contents

Module 1: Everyone

1. security overview and patching
2. public vulnerability databases
3. secure design principles and process
4. security assessment and testing
5. shell and environment
6. resource exhaustion
7. trust management

Module 2: Developers

1. Buffer overflows
2. Format strings
3. Input validation
4. Cross-site scripting vulnerabilities
5. Symlinks and race conditions
6. Temporary folders and random numbers
7. Canonicalization and directory traversal

Module 3: Network Application Developers & Architects

1. Networking architecture
2. Physical and link layers
3. Network layer
4. Transport layer
5. DNS, RPC, NFS
6. Routing
7. Wireless networks
8. IPv6 and xSEC protocol extensions

Rarely Addressed Windows Threats

During the course of adapting symlink vulnerability material to Windows, we realized that NFS directory junctions (reparse points) would allow the equivalent of symlink attacks to be mounted. Moreover, the Windows API calls make this very difficult to detect. However, this problem has not been addressed in any material that we have seen. We propose a solution, which we call the Windows directory crawl. In order to make sure that such an attack can't be mounted, directories must be opened in order from Drive: down to the destination, and handles must be kept open to all the directories in a Windows path until the operations are completed. We also noted the difficulties of operating in shared directories under Windows. An alternative is to operate only in trusted, secured directories.

Thanks to Alan Krassowski for verifying exploitation.

Impact

- Material used in several other institutions (with modifications)
- Over 19 200 .ppt external downloads since release
  (Purdue downloads were subtracted based on the IP address; each download is a set of slides; there are several sets of slides per module)
- 4337 .pdf downloads
- Extrapolates to 43 500 downloads/year

Future Plans

- Need to disperse and try out the material at different institutions and companies
- Need to add consideration of session fixation attacks

PURDUE UNIVERSITY

CERIAS®

Discovery Park e-Enterprise Center