Security in Software Engineering

Web Services
Part 11 : Good practices for Secure Software.

Alexandre Dulaunoy

adulau@foo.be

Agenda

- A little story,
- Buffer Overflows,
- Race Conditions,
- Security in Operations,
- Conclusion.

Buffer Overflows

- Process/Programs create sections in memory (heap, stack),
- A buffer is various "continuous" chunks of memory,
- What happens if the continuous chunks is overwritten ?,
  - Programs can act act in strange ways,
  - Programs can fail completely,
  - Programs can proceed without any noticeable difference
- The little story of the 5th Dec (brk() syscall exploit in Linux kernel).

brk() syscall exploit in Linux Kernel

- The kernel organizes a process's memory in a specific structure,
- The kernel is also a program that "controls" all the programs on the system,
- brk() is used to extend heap memory when requested,
- Specific design on IA-32 systems (the kernel is mapped after the 0xc0000000),
- This exploit is working locally (not remotely) but ...,

Buffer Overflows - how to avoid ?

- Make defensive programming,
- Avoid a lot of default call (like gets() for libc, system.* in java...),
- Use source-code scanning tools whenever possible,
- Compile with stackguard-like system,
- Use on the running system a protection against buffer overflow,
- Don’t rely on one system protection.

Race Conditions

- The example of the meeting,
- RC occurs when an assumption needs to hold true for a period of time but actually may not,
- Race condition has a window of vulnerability,
- On computer systems, the window of vulnerability is often quite small,
- File-based race conditions are the most important,
- More related to multi-users systems (like Unix),

Race Conditions : example

- class Counter example (via java.servlet) (the "synchronized(obj)" solution),
- race condition in passwd (Solaris example),
- Payment system and race conditions (ATM system example),
- Asynchronous message delivery and race conditions (another payment system example),
- and the issue of the 5th Dec...

Race Conditions - how to avoid ?

- Difficult to find and debug,
- Apply rules of concurrency problem,
- Limit usage of multithreaded/multiprocess (only when required),
- Think about each microseconds between each operations,
- Be atomic...

Operations to maintain security

- Security is everybody's problem but no one cares,
- Network environment
  - Monitor,
  - Only essential net services,
  - Separate management network from production,
  - Multiple layers of security,
  - Log network events (and monitor...),

Deploy application with security in mind

- Secure the Operating System (don’t trust the environment),
- Third-party code should be fully audited,
- Ensure daily operations practices,
- Be ready for worst case situation,
- and don’t forget the issue of the 5th Dec...

Q&R

- adulau@foo.be
- http://www.foo.be/cours/securite-webservices/
- 3B12 DCC2 82FA 2931 2F5B 709A 09E2 CD49 44E6 CBCD