ISO/IEC JTC 1/SC 22/OWGV N 0101

OWG: Vulnerability, presentation to conference associated with SC 22 plenary, 28 September 2007

Date	15 October 2007
Contributed by	John Benito
Original file name	owg_v-Singapore_V2.pdf
Notes	

A new type of Working Group used for a new SC22 Working Group

OWG: Vulnerability

John Benito

JTC 1/SC 22 WG14 Convener
INCITS CT 22 Vice Chairman
JTC 1/SC 22 OWG:V Convener

The Problem

- Any programming language has constructs that are imperfectly defined, implementation dependent or difficult to use correctly.

- As a result, software programs sometimes execute differently than intended by the writer.

- In some cases, these vulnerabilities can be exploited by hostile parties.
    - – Can compromise safety, security and privacy.
    - – Can be used to make additional attacks.

Complicating Factors

- The choice of programming language for a project is not solely a technical decision and is not made solely by software engineers.

- Some vulnerabilities cannot be mitigated by better use of the language but require mitigation by other methods, e.g. review, static analysis.

An example

- While buffer overflow examples can be rather complex, it is possible to have very simple, yet still exploitable, stack based buffer overflows:

- An Example in the C programming language:

```
#define BUFSIZE 256

int main(int argc, char **argv) {
   char buf[BUFSIZE];

   strcpy(buf, argv[1]);
}
```

Example

- Buffer overflows generally lead to the application halting or crashing.

- Other attacks leading to lack of availability are possible, that can include putting the program into an infinite loop.

- Buffer overflows often can be used to execute arbitrary code, which is usually outside the scope of a program's implicit security policy.

OWG: Vulnerability Status

- Response to NP Ballot comments is completed, see SC 22 N4027

- Project is organized and on schedule to produce a document in 2009

- Current draft is ready for it's first SC 22 ballot

- The project has two officers
  - – Convener/Project Editor, John Benito
  - – Secretary, Jim Moore

OWG: Vulnerability Status

- Five meetings have been held, hosted by
  - US
  - Italy
  - Canada
  - UK
- Meetings planned through 2008, hosted by
  - Netherlands
  - US
  - Germany
- E-Mail reflector, Wiki and Web site are used during and between meetings
- More information
  - http://aitc.aitcnet.org/isai/

OWG: Vulnerability Status

- The body of Technical Report describes vulnerabilities in a generic manner, including:
  - Brief description of application vulnerability
  - Cross-reference to enumerations, e.g. CWE
  - Categorizations by selected characteristics
  - Description of failure mechanism, i.e. how coding problem relates to application vulnerability
  - Points at which the causal chain could be broken
  - Assumed variations among languages
  - Ways to avoid the vulnerability or mitigate its effects
- Annexes will provide language-specific treatments of each vulnerability.

Meeting Schedule for OWG:V

- Meeting #6 2007-10-1/3 INCITS/Plum Hall, Kona, Hawaii, USA

- Meeting #7 2007-12-12/14 INCITS/SEI, Pittsburgh, PA, USA

- Meeting #8 2008-04-09/11 NEN/ACE, Amsterdam, NL

- Meeting #9 2008-07 INCITS/Blue Pilot, Washington DC, USA

- Meeting #10 2008-10 – Stuttgart, Germany

OWG: Vulnerability Participants

- Canada
- Germany
- Italy
- Japan
- France
- United Kingdom
- USA – CT 22
- SC 22/WG 9
- SC 22/WG14
- MDC (Mumps)
- SC 22/WG 5, INCITS J3 (Fortran)
- SC 22/WG 4, INCITS J4 (Cobol)
- ECMA (C#, C++CLI)
- RT/SC Java
- MISRA C/C++
- CERT

OWG:Vulnerability Progress

- A document suitable for registration has been completed.

- A template for vulnerability descriptions has been completed.

- An initial set of vulnerabilities has been proposed for treatment.

OWG:Vulnerability Product

- ## A type III Technical Report
  - A document containing information of a different kind from that which is normally published as an International Standard

- ## Project is to work on a set of *common mode* failures that occur across a variety of languages
  - Not all vulnerabilities are common to all languages, that is, some manifest in just a language

- ## The product will not contain *normative* statements, but information and suggestions

OWG:Vulnerability Product

- No single programming language or family of programming languages is to be singled out

  - As many programming languages as possible should be involved

  - Need not be just the languages defined by ISO Standards

Approach to Identifying Vulnerabilities

- *Empirical approach:* Observe the vulnerabilities that occur in the wild and describe them, e.g. buffer overrun, execution of unvalidated remote content

- *Analytical approach:* Identify potential vulnerabilities through analysis of programming languages
  - This just might help in identifying tomorrows vulnerabilities.

Audience

- *Safety*: Products where it is critical to prevent behavior which might lead to human injury, and it is justified to spend additional development money

- *Security*: Products where it is critical to secure data or access, and it is justified to spend additional development money

- *Predictability*: Products where high confidence in the result of the computation is desired

- *Assurance*: Products to be developed for dependability or other important characteristics

Measure of Success

- Provide guidance to users of programming languages that:
    - Assists them in improving the predictability of the execution of their software even in the presence of an attacker
    - Informs their selection of an appropriate programming language for their job
- Provide feedback to programming language standardization groups, resulting in the improvement of programming language standards.

OWG: Vulnerability Summary

- ## We are making progress!
  - ❑ meetings scheduled out over a year
  - ❑ Participation is good and is made up of a wide variety of technical expertise.

- ## Have a document that is ready for the first SC 22 ballot (registration).

- ## On track to publish in 2009.