# CWE漏洞文档

1. **Insufficient Control Flow Management**

|  |  |
| --- | --- |
| 英文名 | Unintended Reentrant Invocation of Non-reentrant Code Via Nested Calls-（1265） |
| 英文描述 | In complex software, a single function call may lead to many different possible code paths, some of which may involve deeply nested calls. It may be difficult to foresee all possible code paths that could emanate from a given function call. In some systems, an external actor can manipulate inputs to the system and thereby achieve a wide range of possible control flows. This is frequently of concern in software that executes script from untrusted sources. Examples of such software are web browsers and PDF readers. A weakness is present when one of the possible code paths resulting from a function call alters program state that the original caller assumes to be unchanged during the call. |
| 中文名 | 不可重入代码通过嵌套调用产生了意外的重入调用 |
| 中文描述 | 在复杂的软件中，单个函数调用可能导致不同的代码路径，其中一些路径也许会包含深度的嵌套调用。对于一个函数的调用，很难预见它在执行过程中可能会产生的代码路径。在一些系统中，一个外部的调用者可以操控系统的输入，从而实现各种可能的控制流。在执行那些不受信任的来源的脚本时，这通常是很令人担忧的。例如web浏览器和PDF阅读器这样的软件。当由函数调用产生的可能代码路径之一更改了原始调用者认为在调用过程中不会更改的程序状态时，这就是一个薄弱点。 |
| 理解情况 | 从示例代码来看，这段代码的能看到的漏洞在于调用了一个不知道来源的脚本，而这个脚本是可能产生攻击性的，比如这个脚本恶意调用**changeBackgroundImage（）**就可以删掉原来的背景图片。 |
| 备注 |  |
| 链接 | https://cwe.mitre.org/data/definitions/1265.html |

**示例代码：**

**class Widget**

**{**

**private:**

**Image\* backgroundImage;**

**public:**

**void click()**

**{**

**if (backgroundImage)**

**{**

**backgroundImage->click();**

**}**

**}**

**void changeBackgroundImage(Image\* newImage)**

**{**

**if (backgroundImage)**

**{**

**delete backgroundImage;**

**}**

**backgroundImage = newImage;**

**}**

**}**

**class Image**

**{**

**public:**

**void click()**

**{**

**scriptEngine->fireOnImageClick();**

**/\* perform some operations using “this” pointer \*/**

**}**

**}**

|  |  |
| --- | --- |
| 英文名 | Sequence of Processor Instructions Leads to Unexpected Behavior (Halt and Catch Fire)-（1281） |
| 英文描述 | If the instruction set architecture (ISA) and processor logic are not designed carefully, and tested thoroughly, certain combinations of instructions may lead to locking the processor or other unexpected and undesirable behavior. Upon encountering unimplemented instruction opcodes or illegal instruction operands the processor should throw an exception and carry on without negatively impacting security. However, specific combinations of legal and illegal instructions may cause unexpected behavior with security implications such as allowing unprivileged programs to completely lock the CPU.  Some examples are the Pentium f00f bug, MC6800 HCF, the Cyrix comma bug, and more generally other "Halt and Catch Fire" instructions. |
| 中文名 | 处理器指令序列导致的异常行为（“停止并着火”） |
| 中文描述 | 如果指令集架构和处理器逻辑设计不周密，并且没有经过全面测试，一些特定组合的指令就可能导致处理器锁死或者其他意外的不良行为。在遇到未实现的指令操作码或者非法指令操作数的时候，处理器应该抛出异常，继续运行而不会对安全性产生负面影响。然而，特定的合法和非法指令的组合也可能引起意料行为，并带有安全隐患，例如允许无特权的程序完全锁死CPU。  比如Pentium f00f错误，MC6800HCF，Cyrix逗号错误，以及其他更常见的“停止并着火”指令。 |
| 理解情况 | 指令组合的一些错误，比如某顺序为A指令接着是B指令，但是如果反过来就可能出现错误，比如B指令需要A指令的结果。  一个不太恰当的例子是流水线，假如流水线没有冒险处理机制，就会导致这个类型的错误。 |
| 备注 | 停止并着火的意思是导致计算机停止有效操作。 |
| 链接 | https://cwe.mitre.org/data/definitions/1281.html |