CWE翻译计划

CWE翻译计划 v1.0 by *UESTC* **418**

# 参与贡献人员：

Mardan & Szg(shang Cr7-joker) & LJIJCJ & Michael Tan

Weakness ID: 5

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> J2EE Misconfiguration: Data Transmission Without Encryption zh: -->J2EE配置错误：没有加密的数据传输*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

通过网络发送的信息在传输过程中可能会受到影响。如果数据以明文发送或者是弱加密，则攻击者可能能够读取或修改内容。

### Description:

Information sent over a network can be compromised while in transit. An attacker may be able to read or modify the contents if the data are sent in plaintext or are weakly encrypted.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

如果应用程序使用SSL来保证与客户端浏览器的机密通信，则应用程序配置应该使得无法在没有SSL的情况下查看任何访问受控页面。绕过SSL有三种常见方法：用户手动输入URL并键入“HTTP”而不是“HTTPS”。攻击者故意将用户发送到不安全的URL。程序员错误地创建了应用程序中页面的相对链接，该页面不会从HTTP切换到HTTPS。 （当链接在网站上的公共区域和安全区域之间移动时，这尤其容易。）

Weakness ID: 6

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> J2EE Misconfiguration: Insufficient Session-ID Length zh: -->J2EE配置错误：会话ID长度不足*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

J2EE应用程序配置为使用不足的会话ID长度。

### Description:

The J2EE application is configured to use an insufficient session ID length.

### 详细描述:

如果攻击者可以猜测或窃取会话ID，那么他们就可以接管用户的会话（称为会话劫持）。可能的会话ID的数量随着会话ID长度的增加而增加，使得猜测或窃取会话ID变得更加困难。

### Extended Description:

If an attacker can guess or steal a session ID, then they may be able to take over the user's session (called session hijacking). The number of possible session IDs increases with increased session ID length, making it more difficult to guess or steal a session ID.

### 问题背景 (Background Detail):

会话ID可用于识别Web环境中的通信方。猜测有效会话标识符所需的预期秒数由以下等式给出：（2 ^ B + 1）/（2 \* A \* S）其中： - B是会话标识符中的熵的比特数。 - A是攻击者每秒可以尝试的猜测次数。 - S是有效会话标识符的数量，这些标识符在任何给定时间都是有效且可以猜到的。会话标识符中的熵比特数总是小于会话标识符中的比特总数。例如，如果会话标识符按升序提供，则无论标识符的长度如何，会话标识符中的熵都将接近零比特。假设使用良好的随机数源生成会话标识符，我们将估计会话标识符中的熵的比特数是会话标识符中的总比特数的一半。对于实际的标识符长度，这是可能的，尽管可能是乐观的。

### 笔记 (Notes):

没有笔记

Weakness ID: 7

提交日期 2009-03-10---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> J2EE Misconfiguration: Missing Custom Error Page zh: -->J2EE配置错误：缺少自定义错误页面*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序的默认错误页面不应显示有关软件系统的敏感信息。

### Description:

The default error page of a web application should not display sensitive information about the software system.

### 详细描述:

Web应用程序必须为4xx错误（例如404），5xx（例如500）错误定义默认错误页面并捕获java.lang.Throwable异常以防止攻击者从应用程序容器的内置错误响应中挖掘信息。  
当攻击者探索寻找漏洞的网站时，该网站提供的信息量对于任何企图攻击的最终成功或失败至关重要。

### Extended Description:

A Web application must define a default error page for 4xx errors (e.g. 404), 5xx (e.g. 500) errors and catch java.lang.Throwable exceptions to prevent attackers from mining information from the application container's built-in error response.  
When an attacker explores a web site looking for vulnerabilities, the amount of information that the site provides is crucial to the eventual success or failure of any attempted attacks.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 8

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> J2EE Misconfiguration: Entity Bean Declared Remote zh: -->J2EE配置错误：实体Bean声明为远程*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当应用程序公开实体bean的远程接口时，它还可能公开获取或设置bean数据的方法。可以利用这些方法来读取敏感信息，或以违反应用程序期望的方式更改数据，从而可能导致其他漏洞。

### Description:

When an application exposes a remote interface for an entity bean, it might also expose methods that get or set the bean's data. These methods could be leveraged to read sensitive information, or to change data in ways that violate the application's expectations, potentially leading to other vulnerabilities.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

暴露远程接口的实体bean成为应用程序攻击面的一部分。出于性能原因，应用程序应该很少使用远程实体bean，因此远程实体bean声明很可能是一个错误。

Weakness ID: 9

提交日期 2008-04-11---> 修改日期 2014-06-23

* **Weakness Name:** *en: --> J2EE Misconfiguration: Weak Access Permissions for EJB Methods zh: -->J2EE配置错误：EJB方法的弱访问权限*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果将提升的访问权限分配给EJB方法，则攻击者可以利用这些权限来利用软件系统。

### Description:

If elevated access rights are assigned to EJB methods, then an attacker can take advantage of the permissions to exploit the software system.

### 详细描述:

如果EJB部署描述符包含一个或多个授予对特殊ANYONE角色的访问权限的方法权限，则表示尚未完全考虑应用程序的访问控制，或者应用程序的结构是以合理的访问控制限制为不可能。

### Extended Description:

If the EJB deployment descriptor contains one or more method permissions that grant access to the special ANYONE role, it indicates that access control for the application has not been fully thought through or that the application is structured in such a way that reasonable access control restrictions are impossible.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 11

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> ASP.NET Misconfiguration: Creating Debug Binary zh: -->ASP.NET配置错误：创建调试二进制文件*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

调试消息可帮助攻击者了解系统并规划一种攻击形式。

### Description:

Debugging messages help attackers learn about the system and plan a form of attack.

### 详细描述:

可以将ASP .NET应用程序配置为生成调试二进制文件。这些二进制文件提供详细的调试消息，不应在生产环境中使用。调试二进制文件旨在用于开发或测试环境，如果将它们部署到生产环境中，则可能会带来安全风险。

### Extended Description:

ASP .NET applications can be configured to produce debug binaries. These binaries give detailed debugging messages and should not be used in production environments. Debug binaries are meant to be used in a development or testing environment and can pose a security risk if they are deployed to production.

### 问题背景 (Background Detail):

<compilation>标记的debug属性定义编译的二进制文件是否应包含调试信息。调试二进制文件的使用使应用程序向用户提供尽可能多的关于自身的信息。

### 笔记 (Notes):

没有笔记

Weakness ID: 12

提交日期 2009-03-10---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> ASP.NET Misconfiguration: Missing Custom Error Page zh: -->ASP.NET配置错误：缺少自定义错误页面*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

ASP .NET应用程序必须启用自定义错误页面，以防止攻击者从框架的内置响应中挖掘信息。

### Description:

An ASP .NET application must enable custom error pages in order to prevent attackers from mining information from the framework's built-in responses.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

<customErrors>标记的mode属性定义是使用自定义错误页面还是默认错误页面。

### 笔记 (Notes):

没有笔记

Weakness ID: 13

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> ASP.NET Misconfiguration: Password in Configuration File zh: -->ASP.NET配置错误：配置文件中的密码*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在配置文件中存储明文密码允许任何能够读取文件的人访问受密码保护的资源，使其成为攻击者的轻松目标。

### Description:

Storing a plaintext password in a configuration file allows anyone who can read the file access to the password-protected resource making them an easy target for attackers.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 14

提交日期 2008-04-11---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Compiler Removal of Code to Clear Buffers zh: -->编译器删除代码以清除缓冲区*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

根据源代码清除敏感内存，但编译器优化会在不再次读取内存时保持内存不变，即“死存储删除”。

### Description:

Sensitive memory is cleared according to the source code, but compiler optimizations leave the memory untouched when it is not read from again, aka "dead store removal."

### 详细描述:

发生以下编译器优化错误：  
  
1.秘密数据存储在内存中。  
2.通过覆盖其内容从内存中清除秘密数据。  
3.使用优化编译器编译源代码，该编译器识别并删除将内容覆盖为死存储的函数，因为随后不使用该存储器。

### Extended Description:

This compiler optimization error occurs when:  
  
1. Secret data are stored in memory.  
2. The secret data are scrubbed from memory by overwriting its contents.  
3. The source code is compiled using an optimizing compiler, which identifies and removes the function that overwrites the contents as a dead store because the memory is not used subsequently.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 15

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> External Control of System or Configuration Setting zh: -->系统外部控制或配置设置*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

一个或多个系统设置或配置元素可以由用户外部控制。

### Description:

One or more system settings or configuration elements can be externally controlled by a user.

### 详细描述:

允许外部控制系统设置可能会中断服务或导致应用程序以意外和潜在的恶意方式运行。

### Extended Description:

Allowing external control of system settings can disrupt service or cause an application to behave in unexpected, and potentially malicious ways.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 20

提交日期 2009-01-12---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Input Validation zh: -->输入验证不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Usable*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品不验证或错误验证可能影响程序的控制流或数据流的输入。

### Description:

The product does not validate or incorrectly validates input that can affect the control flow or data flow of a program.

### 详细描述:

当软件未正确验证输入时，攻击者能够以应用程序其余部分不期望的形式制作输入。这将导致系统的某些部分接收到非预期的输入，这可能导致控制流的改变，资源的任意控制或任意代码执行。

### Extended Description:

When software does not validate input properly, an attacker is able to craft the input in a form that is not expected by the rest of the application. This will lead to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

CWE-116和CWE-20具有紧密关联，因为根据结构化消息的性质，正确的输入验证可以间接防止特殊字符改变结构化消息的含义。例如，通过验证数字ID字段应仅包含0-9个字符，程序员可以有效地防止注入攻击。但是，输入验证并不总是足够的，尤其是在必须支持不太严格的数据类型时，例如自由格式文本。考虑一个SQL注入场景，其中将姓氏插入到查询中。名称“O'Reilly”可能会通过验证步骤，因为它是英语中常见的姓氏。但是，它不能直接插入数据库，因为它包含“'”撇号字符，需要转义或以其他方式中和。在这种情况下，剥离撇号可能会降低SQL注入的风险，但它会产生不正确的行为，因为会记录错误的名称。

Weakness ID: 22

提交日期 2010-02-16---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') zh: -->限制目录的路径名的不正确限制（“路径遍历”）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造路径名，该路径名用于标识位于受限父目录下的文件或目录，但该软件未正确中和路径名中可能导致路径名解析到的位置的特殊元素在受限制的目录之外。

### Description:

The software uses external input to construct a pathname that is intended to identify a file or directory that is located underneath a restricted parent directory, but the software does not properly neutralize special elements within the pathname that can cause the pathname to resolve to a location that is outside of the restricted directory.

### 详细描述:

许多文件操作都是在受限目录中进行的。通过使用诸如“..”和“/”分隔符之类的特殊元素，攻击者可以在受限制的位置之外逃逸，以访问系统中其他位置的文件或目录。最常见的特殊元素之一是“../”序列，在大多数现代操作系统中，它被解释为当前位置的父目录。这被称为相对路径遍历。路径遍历还包括使用绝对路径名，例如“/ usr / local / bin”，这在访问意外文件时也很有用。这被称为绝对路径遍历。  
在许多编程语言中，注入空字节（0或NUL）可能允许攻击者截断生成的文件名以扩大攻击范围。例如，软件可以将“.txt”添加到任何路径名，从而将攻击者限制为文本文件，但空注入可以有效地消除此限制。

### Extended Description:

Many file operations are intended to take place within a restricted directory. By using special elements such as ".." and "/" separators, attackers can escape outside of the restricted location to access files or directories that are elsewhere on the system. One of the most common special elements is the "../" sequence, which in most modern operating systems is interpreted as the parent directory of the current location. This is referred to as relative path traversal. Path traversal also covers the use of absolute pathnames such as "/usr/local/bin", which may also be useful in accessing unexpected files. This is referred to as absolute path traversal.  
In many programming languages, the injection of a null byte (the 0 or NUL) may allow an attacker to truncate a generated filename to widen the scope of attack. For example, the software may add ".txt" to any pathname, thus limiting the attacker to text files, but a null injection may effectively remove this restriction.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

路径名等价可以视为一种规范化错误。

Weakness ID: 23

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Relative Path Traversal zh: -->相对路径遍历*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造应该在受限目录内的路径名，但它不能正确地中和可以解析到该目录之外的位置的序列，例如“..”。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize sequences such as ".." that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 24

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '../filedir' zh: -->路径遍历：'../filedir'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造应该在受限目录内的路径名，但它不能正确地中和可以解析到该目录之外的位置的“../”序列。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize "../" sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
“../”操作是对使用“/”作为目录分隔符的操作系统的规范操作，例如基于UNIX和Linux的系统。在某些情况下，在支持“/”但不支持主分隔符的环境中绕过保护方案很有用，例如Windows，它使用“\”但也可以接受“/”。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
The "../" manipulation is the canonical manipulation for operating systems that use "/" as directory separators, such as UNIX- and Linux-based systems. In some cases, it is useful for bypassing protection schemes in environments for which "/" is supported but not the primary separator, such as Windows, which uses "\" but can also accept "/".

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 25

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '/../filedir' zh: -->路径遍历：'/../filedir'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造应该在受限目录内的路径名，但它不能正确地中和可以解析到该目录之外的位置的“/../”序列。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize "/../" sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
有时程序会在输入开头检查“../”，因此“/../”可以绕过该检查。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
Sometimes a program checks for "../" at the beginning of the input, so a "/../" can bypass that check.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 26

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '/dir/../filename' zh: -->路径遍历：'/ dir /../filename'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造应该在受限目录内的路径名，但它不能正确地中和可以解析到该目录之外的位置的“/dir/../filename”序列。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize "/dir/../filename" sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
'/dir/../filename'操作对于绕过某些路径遍历保护方案很有用。有时程序只在输入开头检查“../”，因此“/../”可以绕过该检查。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
The '/dir/../filename' manipulation is useful for bypassing some path traversal protection schemes. Sometimes a program only checks for "../" at the beginning of the input, so a "/../" can bypass that check.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 27

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: 'dir/../../filename' zh: -->路径遍历：'dir /../../ filename'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造应该在受限目录内的路径名，但它不能正确地中和可以解析到该目录之外的位置的多个内部“../”序列。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize multiple internal "../" sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
'目录/../../ filename'操作对于绕过某些路径遍历保护方案很有用。有时程序只删除一个“../”序列，因此多个“../”可以绕过该检查。或者，此操作可用于绕过路径名开头的“../”检查，向上移动多个目录级别。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
The 'directory/../../filename' manipulation is useful for bypassing some path traversal protection schemes. Sometimes a program only removes one "../" sequence, so multiple "../" can bypass that check. Alternately, this manipulation could be used to bypass a check for "../" at the beginning of the pathname, moving up more than one directory level.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 28

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '..\filedir' zh: -->路径遍历：'.. \ filedir'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造应该在受限目录内的路径名，但它不能正确地中和可以解析到该目录之外的位置的“..”序列。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize "..\" sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
'.. \'操作是对使用“\”作为目录分隔符（如Windows）的操作系统的规范操作。但是，它也可用于绕过仅假设“/”分隔符有效的路径遍历保护方案。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
The '..\' manipulation is the canonical manipulation for operating systems that use "\" as directory separators, such as Windows. However, it is also useful for bypassing path traversal protection schemes that only assume that the "/" separator is valid.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 29

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '\..\filename' zh: -->路径遍历：'\ .. \ filename'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造应该在受限目录内的路径名，但它不能正确地中和可以解析到该目录之外的位置的'\ .. \ filename'（前导反斜点点）序列。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize '\..\filename' (leading backslash dot dot) sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
这类似于CWE-25，除了使用“\”而不是“/”。有时程序会在输入的开头检查“.. \”，因此“\ .. \”可以绕过该检查。它也可用于绕过仅假设“/”分隔符有效的路径遍历保护方案。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
This is similar to CWE-25, except using "\" instead of "/". Sometimes a program checks for "..\" at the beginning of the input, so a "\..\" can bypass that check. It is also useful for bypassing path traversal protection schemes that only assume that the "/" separator is valid.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 30

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '\dir\..\filename' zh: -->路径遍历：'\ dir \ .. \ filename'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造一个应该在受限目录内的路径名，但是它没有正确地中和'\ dir \ .. \ filename'（前导反斜点点）序列，这些序列可以解析到该区域之外的位置目录。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize '\dir\..\filename' (leading backslash dot dot) sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
这类似于CWE-26，除了使用“\”而不是“/”。 '\ dir \ .. \ filename'操作对于绕过某些路径遍历保护方案很有用。有时程序只在输入的开头检查“.. \”，因此“\ .. \”可以绕过该检查。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
This is similar to CWE-26, except using "\" instead of "/". The '\dir\..\filename' manipulation is useful for bypassing some path traversal protection schemes. Sometimes a program only checks for "..\" at the beginning of the input, so a "\..\" can bypass that check.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 31

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: 'dir\..\..\filename' zh: -->路径遍历：'dir \ .. \ .. \ filename'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造一个应该在受限目录内的路径名，但是它没有正确地中和'dir \ .. \ .. \ filename'（多个内部反斜杠点点）序列，这些序列可以解析为一个位置。在该目录之外。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize 'dir\..\..\filename' (multiple internal backslash dot dot) sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
'dir \ .. \ .. \ filename'操作对于绕过某些路径遍历保护方案很有用。有时程序只删除一个“..”序列，因此多个“.. \”可以绕过该检查。或者，此操作可用于绕过路径名开头的“.. \”检查，向上移动多个目录级别。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
The 'dir\..\..\filename' manipulation is useful for bypassing some path traversal protection schemes. Sometimes a program only removes one "..\" sequence, so multiple "..\" can bypass that check. Alternately, this manipulation could be used to bypass a check for "..\" at the beginning of the pathname, moving up more than one directory level.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 32

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '...' (Triple Dot) zh: -->路径遍历：'...'（三点）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造应该在受限目录内的路径名，但它不能正确地中和可以解析到该目录之外的位置的“...”（三点）序列。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize '...' (triple dot) sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
“...”操作对于绕过某些路径遍历保护方案很有用。在某些Windows系统上，它等同于“.. \ ..”并可能绕过假设只有两个点有效的检查。不完整的过滤，例如去除“./”序列，最终会产生有效的“..”序列，因为它们会陷入不安全的值（CWE-182）。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
The '...' manipulation is useful for bypassing some path traversal protection schemes. On some Windows systems, it is equivalent to "..\.." and might bypass checks that assume only two dots are valid. Incomplete filtering, such as removal of "./" sequences, can ultimately produce valid ".." sequences due to a collapse into unsafe value (CWE-182).

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这个以操作为中心的条目目前隐藏着两个截然不同的弱点，因此可能需要拆分。该操作在两种不同的上下文中有效：它在Windows上等同于“.. \ ..”，或者它可以利用不完整的过滤，例如如果程序员在字符串中单次删除“./”（将数据折叠成不安全的值，CWE-182）。

Weakness ID: 33

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '....' (Multiple Dot) zh: -->路径遍历：'....'（多点）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造应该在受限目录内的路径名，但它不能正确地中和可以解析到该目录之外的位置的“......”（多点）序列。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize '....' (multiple dot) sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
'....'操作对于绕过某些路径遍历保护方案很有用。在某些Windows系统上，它相当于“.. \ .. \ ..”并可能绕过假设只有两个点有效的检查。不完整的过滤，例如去除“./”序列，最终会产生有效的“..”序列，因为它们会陷入不安全的值（CWE-182）。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
The '....' manipulation is useful for bypassing some path traversal protection schemes. On some Windows systems, it is equivalent to "..\..\.." and might bypass checks that assume only two dots are valid. Incomplete filtering, such as removal of "./" sequences, can ultimately produce valid ".." sequences due to a collapse into unsafe value (CWE-182).

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

与三点CWE-32一样，这种操作可能隐藏了应该更加明确的多个弱点。

Weakness ID: 34

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '....//' zh: -->路径遍历：'.... //'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造一个应该在受限目录内的路径名，但是它没有正确地中和可以解析到该目录之外的位置的“.... //”（加倍点点斜杠）序列。 。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize '....//' (doubled dot dot slash) sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
'.... //'操作对于绕过某些路径遍历保护方案很有用。如果“../”按顺序方式过滤，如某些正则表达式引擎所做，则“.... //”可以折叠为“../”不安全值（CWE-182）。如果操作系统将“//”和“/”视为等效，则删除“..”时也可能有用。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
The '....//' manipulation is useful for bypassing some path traversal protection schemes. If "../" is filtered in a sequential fashion, as done by some regular expression engines, then "....//" can collapse into the "../" unsafe value (CWE-182). It could also be useful when ".." is removed, if the operating system treats "//" and "/" as equivalent.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能是由于清除错误导致从“.... //”中删除单个“../”

Weakness ID: 35

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '.../...//' zh: -->路径遍历：'... / ... //'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造一个应该在受限目录内的路径名，但是它没有正确地中和可以解析到外部位置的'... / ... //'（加倍的三点斜杠）序列那个目录。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize '.../...//' (doubled triple dot slash) sequences that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。  
'... / ... //'操作对于绕过某些路径遍历保护方案很有用。如果“../”按顺序方式过滤，如某些正则表达式引擎所做，则“... / ... //”可以折叠为“../”不安全值（CWE-182）。删除第一个“../”会产生“.... //”;第二次删除产生“../”。根据算法，软件可能对CWE-34敏感，但对CWE-35不敏感，反之亦然。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.  
The '.../...//' manipulation is useful for bypassing some path traversal protection schemes. If "../" is filtered in a sequential fashion, as done by some regular expression engines, then ".../...//" can collapse into the "../" unsafe value (CWE-182). Removing the first "../" yields "....//"; the second removal yields "../". Depending on the algorithm, the software could be susceptible to CWE-34 but not CWE-35, or vice versa.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 36

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Absolute Path Traversal zh: -->绝对路径遍历*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入来构造应该在受限目录内的路径名，但它不能正确地中和绝对路径序列，例如“/ abs / path”，它可以解析到该目录之外的位置。

### Description:

The software uses external input to construct a pathname that should be within a restricted directory, but it does not properly neutralize absolute path sequences such as "/abs/path" that can resolve to a location that is outside of that directory.

### 详细描述:

这允许攻击者遍历文件系统以访问受限目录之外的文件或目录。

### Extended Description:

This allows attackers to traverse the file system to access files or directories that are outside of the restricted directory.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 37

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '/absolute/pathname/here' zh: -->路径遍历：'/ absolute / pathname / here'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下以斜杠绝对路径（'/ absolute / pathname / here'）的形式接受输入的软件系统可以允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts input in the form of a slash absolute path ('/absolute/pathname/here') without appropriate validation can allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 38

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '\absolute\pathname\here' zh: -->路径遍历：'\ absolute \ pathname \ here'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有经过适当验证的情况下以反斜杠绝对路径（'\ absolute \ pathname \ here'）的形式接受输入的软件系统可以允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts input in the form of a backslash absolute path ('\absolute\pathname\here') without appropriate validation can allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 39

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: 'C:dirname' zh: -->路径遍历：'C：dirname'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

攻击者可以将驱动器号或Windows卷号（“C：dirname”）注入软件系统，以潜在地重定向对非预期位置或任意文件的访问。

### Description:

An attacker can inject a drive letter or Windows volume letter ('C:dirname') into a software system to potentially redirect access to an unintended location or arbitrary file.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 40

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Traversal: '\\UNC\share\name\' (Windows UNC Share) zh: -->路径遍历：'\\ UNC \ share \ name \'（Windows UNC Share）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

攻击者可以将Windows UNC共享（'\\ UNC \ share \ name'）注入软件系统，以潜在地重定向对非预期位置或任意文件的访问。

### Description:

An attacker can inject a Windows UNC share ('\\UNC\share\name') into a software system to potentially redirect access to an unintended location or arbitrary file.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 41

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Resolution of Path Equivalence zh: -->路径等价的不正确解决方案*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

系统或应用程序容易受到路径等效的文件系统内容泄露。路径等效涉及在文件和目录名称中使用特殊字符。相关的操作旨在为同一对象生成多个名称。

### Description:

The system or application is vulnerable to file system contents disclosure through path equivalence. Path equivalence involves the use of special characters in file and directory names. The associated manipulations are intended to generate multiple names for the same object.

### 详细描述:

通常采用路径等价来规避使用不完整的文件名或文件路径表示表示的访问控制。这与路径遍历不同，其中执行操作以生成不同对象的名称。

### Extended Description:

Path equivalence is usually employed in order to circumvent access controls expressed using an incomplete set of file name or file path representations. This is different from path traversal, wherein the manipulations are performed to generate a name for a different object.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

其中一些操作也可能在路径遍历问题中有效。

Weakness ID: 42

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: 'filename.' (Trailing Dot) zh: -->路径等价：'文件名。' （尾随点）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下接受尾随点（'filedir。'）形式的路径输入的软件系统可能导致模糊的路径解析，并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of trailing dot ('filedir.') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 43

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: 'filename....' (Multiple Trailing Dot) zh: -->路径等价：'filename ....'（多个尾随点）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下接受多个尾随点（'filedir ....'）形式的路径输入的软件系统可能导致模糊的路径解析，并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of multiple trailing dot ('filedir....') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 44

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: 'file.name' (Internal Dot) zh: -->路径等价：'file.name'（内部点）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下接受内部点（'file.ordir'）形式的路径输入的软件系统可能导致模糊的路径解析，并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of internal dot ('file.ordir') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

从字符串中删除内部点的不正确尝试可能导致CWE-181（错误行为顺序：过滤前验证）。

Weakness ID: 45

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: 'file...name' (Multiple Internal Dot) zh: -->路径等价：'文件...名称'（多个内部点）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下接受多个内部点（'file ... dir'）形式的路径输入的软件系统可能导致模糊的路径解析，并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of multiple internal dot ('file...dir') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

从字符串中删除内部点的不正确尝试可能导致CWE-181（错误行为顺序：过滤前验证）。

Weakness ID: 46

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: 'filename ' (Trailing Space) zh: -->路径等价：'文件名'（尾随空格）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下接受尾随空间（'filedir'）形式的路径输入的软件系统可能导致模糊的路径解析，并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of trailing space ('filedir ') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 47

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: ' filename' (Leading Space) zh: -->路径等价：'文件名'（前导空格）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下接受前导空间（'filedir'）形式的路径输入的软件系统可能导致模糊的路径解析，并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of leading space (' filedir') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 48

提交日期 2008-04-11---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Path Equivalence: 'file name' (Internal Whitespace) zh: -->路径等价：'文件名'（内部空白）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有经过适当验证的情况下接受内部空间（'文件（SPACE）名称'）形式的路径输入的软件系统可能导致模糊的路径解析，并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of internal space ('file(SPACE)name') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这种弱点可能会引用引用问题，例如： “Program Files”不带引号的搜索路径（CWE-428）。如果过滤删除所有无关的空格，它也可能是等价问题。

Weakness ID: 49

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: 'filename/' (Trailing Slash) zh: -->路径等价：'filename /'（尾随斜线）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下以尾部斜杠（'filedir /'）的形式接受路径输入的软件系统可能导致模糊的路径解析，并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of trailing slash ('filedir/') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 50

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: '//multiple/leading/slash' zh: -->路径等价：'// multiple / leading / slash'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下以多个前导斜杠（'// multiple / leading / slash'）的形式接受路径输入的软件系统可能导致模糊的路径解析并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of multiple leading slash ('//multiple/leading/slash') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 51

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: '/multiple//internal/slash' zh: -->路径等价：'/ multiple // internal / slash'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下以多个内部斜杠（'/ multiple // internal / slash /'）的形式接受路径输入的软件系统可能导致模糊路径解析并允许攻击者将文件系统遍历到非预期的位置或任意访问文件。

### Description:

A software system that accepts path input in the form of multiple internal slash ('/multiple//internal/slash/') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 52

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: '/multiple/trailing/slash//' zh: -->路径等价：'/ multiple / trailing / slash //'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下以多个尾部斜杠（'/ multiple / trailing / slash //'）的形式接受路径输入的软件系统可能导致模糊的路径解析并允许攻击者将文件系统遍历到非预期的位置或任意访问文件。

### Description:

A software system that accepts path input in the form of multiple trailing slash ('/multiple/trailing/slash//') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 53

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: '\multiple\\internal\backslash' zh: -->路径等价：'\ multiple \\ internal \ backslash'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有经过适当验证的情况下以多个内部反斜杠（'\ multiple \ trailing \\ slash'）的形式接受路径输入的软件系统可能导致模糊的路径解析并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of multiple internal backslash ('\multiple\trailing\\slash') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 54

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: 'filedir\' (Trailing Backslash) zh: -->路径等价：'filedir'（尾随反斜杠）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当验证的情况下以尾随反斜杠（'filedir \'）的形式接受路径输入的软件系统可能导致模糊的路径解析，并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of trailing backslash ('filedir\') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 55

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: '/./' (Single Dot Directory) zh: -->路径等价：'/。/'（单点目录）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有经过适当验证的情况下接受单点目录漏洞（'/./'）形式的路径输入的软件系统可能导致模糊的路径解析，并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of single dot directory exploit ('/./') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 56

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: 'filedir\*' (Wildcard) zh: -->路径等价：'filedir \*'（通配符）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有经过适当验证的情况下接受星号通配符（'filedir \*'）形式的路径输入的软件系统可能导致模糊的路径解析，并允许攻击者将文件系统遍历到非预期的位置或访问任意文件。

### Description:

A software system that accepts path input in the form of asterisk wildcard ('filedir\*') without appropriate validation can lead to ambiguous path resolution and allow an attacker to traverse the file system to unintended locations or access arbitrary files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 57

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Path Equivalence: 'fakedir/../realdir/filename' zh: -->路径等价：'fakedir /../ realdir / filename'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含限制对“realdir / filename”的访问的保护机制，但它使用“fakedir /../ realdir / filename”形式的外部输入构造路径名，这些输入不由这些机制处理。这允许攻击者对目标文件执行未经授权的操作。

### Description:

The software contains protection mechanisms to restrict access to 'realdir/filename', but it constructs pathnames using external input in the form of 'fakedir/../realdir/filename' that are not handled by those mechanisms. This allows attackers to perform unauthorized actions against the targeted file.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这是一种操作，它使用注入一个结果（使用相对路径的包含违规）来实现不同的结果（通过备用名称等同）。

Weakness ID: 58

提交日期 2008-04-11---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Path Equivalence: Windows 8.3 Filename zh: -->路径等效：Windows 8.3文件名*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个保护机制，限制在Windows操作系统上访问长文件名，但该软件没有正确限制对等效的短“8.3”文件名的访问。

### Description:

The software contains a protection mechanism that restricts access to a long filename on a Windows operating system, but the software does not properly restrict access to the equivalent short "8.3" filename.

### 详细描述:

在以后的Windows操作系统中，文件可以具有“长名称”和与旧Windows文件系统兼容的短名称，文件名最多8个字符，扩展名最多3个字符。因此，这些“8.3”文件名充当具有长名称的文件的备用名称，因此它们是有用的路径名等效操作。

### Extended Description:

On later Windows operating systems, a file can have a "long name" and a short name that is compatible with older Windows file systems, with up to 8 characters in the filename and 3 characters for the extension. These "8.3" filenames, therefore, act as an alternate name for files with long names, so they are useful pathname equivalence manipulations.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可能未充分研究

Weakness ID: 59

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Link Resolution Before File Access ('Link Following') zh: -->文件访问前的链接解析不正确（'链接跟随'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件会尝试根据文件名访问文件，但不能正确阻止该文件名识别解析为非预期资源的链接或快捷方式。

### Description:

The software attempts to access a file based on the filename, but it does not properly prevent that filename from identifying a link or shortcut that resolves to an unintended resource.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

软链接是UNIX术语，与基于Windows的平台上的简单快捷方式同义。

### 笔记 (Notes):

链接以下漏洞是多因素漏洞（MFV）。它们是多个元素的组合：文件或目录权限，文件名可预测性，竞争条件，在某些情况下，还有设计限制，其中没有执行原子文件创建操作的机制。一些潜在因素是竞争条件，权限和可预测性。

Weakness ID: 61

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> UNIX Symbolic Link (Symlink) Following zh: -->UNIX符号链接（符号链接）以下*
* **Abstraction:** *Compound* **Structure:** *Composite* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

打开文件或目录时，该软件无法充分说明文件是何时解析为目标控制范围之外的目标的符号链接。这可能允许攻击者使软件对未经授权的文件进行操作。

### Description:

The software, when opening a file or directory, does not sufficiently account for when the file is a symbolic link that resolves to a target outside of the intended control sphere. This could allow an attacker to cause the software to operate on unauthorized files.

### 详细描述:

允许UNIX符号链接（符号链接）作为路径的一部分的软件系统，无论是在内部代码中还是通过用户输入，都允许攻击者欺骗符号链接并将文件系统遍历到非预期的位置或访问任意文件。符号链接可以允许攻击者读取/写入/损坏他们最初没有访问权限的文件。

### Extended Description:

A software system that allows UNIX symbolic links (symlink) as part of paths whether in internal code or through user input can allow an attacker to spoof the symbolic link and traverse the file system to unintended locations or access arbitrary files. The symbolic link can permit an attacker to read/write/corrupt a file that they originally did not have permissions to access.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在C和shell程序中经常发现Symlink漏洞，但所有编程语言都有这个问题。即使是shell程序也可能报告不足。调用遵循符号链接的其他程序的程序中可能存在“二阶符号链接漏洞”。它们很少报告，但在使用进程调用时可能相当普遍。参考：[Christey2005]

Weakness ID: 62

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> UNIX Hard Link zh: -->UNIX硬链接*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当打开文件或目录时，该软件不足以说明名称何时与目标控制范围之外的目标的硬链接相关联。这可能允许攻击者使软件对未经授权的文件进行操作。

### Description:

The software, when opening a file or directory, does not sufficiently account for when the name is associated with a hard link to a target that is outside of the intended control sphere. This could allow an attacker to cause the software to operate on unauthorized files.

### 详细描述:

系统检查硬链接失败可能导致不同类型的攻击。例如，如果特权程序使用的文件被替换为敏感文件的硬链接（例如/ etc / passwd），则攻击者可以升级其权限。当进程打开文件时，攻击者可以承担该进程的权限。

### Extended Description:

Failure for a system to check for hard links can result in vulnerability to different types of attacks. For example, an attacker can escalate their privileges if a file used by a privileged program is replaced with a hard link to a sensitive file (e.g. /etc/passwd). When the process opens the file, the attacker can assume the privileges of that process.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。检查符号链接的程序很可能容易受到硬链接的攻击。

Weakness ID: 64

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Windows Shortcut Following (.LNK) zh: -->Windows快捷方式（.LNK）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

打开文件或目录时，如果文件是目标位于预期控制范围之外的Windows快捷方式（.LNK），则该软件无法充分处理。这可能允许攻击者使软件对未经授权的文件进行操作。

### Description:

The software, when opening a file or directory, does not sufficiently handle when the file is a Windows shortcut (.LNK) whose target is outside of the intended control sphere. This could allow an attacker to cause the software to operate on unauthorized files.

### 详细描述:

快捷方式（扩展名为.lnk的文件）可以允许攻击者读取/写入他们最初没有访问权限的文件。

### Extended Description:

The shortcut (file with the .lnk extension) can permit an attacker to read/write a file that they originally did not have permissions to access.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。 Windows .LNK文件比Unix符号链接更“可移植”，并且已用于远程攻击。一些Windows API将访问LNK，就像它们是常规文件一样，因此可以预期会更频繁地报告它们。

Weakness ID: 65

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Windows Hard Link zh: -->Windows硬链接*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当打开文件或目录时，该软件在名称与目标控制范围之外的目标的硬链接相关联时无法充分处理。这可能允许攻击者使软件对未经授权的文件进行操作。

### Description:

The software, when opening a file or directory, does not sufficiently handle when the name is associated with a hard link to a target that is outside of the intended control sphere. This could allow an attacker to cause the software to operate on unauthorized files.

### 详细描述:

系统检查硬链接失败可能导致不同类型的攻击。例如，如果特权程序使用的文件被替换为敏感文件的硬链接（例如AUTOEXEC.BAT），则攻击者可以升级其权限。当进程打开文件时，攻击者可以承担该进程的权限，或阻止程序准确处理数据。

### Extended Description:

Failure for a system to check for hard links can result in vulnerability to different types of attacks. For example, an attacker can escalate their privileges if a file used by a privileged program is replaced with a hard link to a sensitive file (e.g. AUTOEXEC.BAT). When the process opens the file, the attacker can assume the privileges of that process, or prevent the program from accurately processing data.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的

Weakness ID: 66

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of File Names that Identify Virtual Resources zh: -->识别虚拟资源的文件名处理不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品不处理或错误处理标识“虚拟”资源的文件名，该文件名未在与文件名关联的目录中直接指定，导致产品对非资源上的资源执行基于文件的操作文件。

### Description:

The product does not handle or incorrectly handles a file name that identifies a "virtual" resource that is not directly specified within the directory that is associated with the file name, causing the product to perform file-based operations on a resource that is not a file.

### 详细描述:

虚拟文件名称表示为普通文件名，但它们实际上是其他资源的别名，这些资源的行为与普通文件不同。根据其功能，它们可以是备用实体。它们不一定列在目录中。

### Extended Description:

Virtual file names are represented like normal file names, but they are effectively aliases for other resources that do not behave like normal files. Depending on their functionality, they could be alternate entities. They are not necessarily listed in directories.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 67

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Handling of Windows Device Names zh: -->Windows设备名称处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件从用户输入构造路径名，但它不处理或错误地处理包含Windows设备名称（如AUX或CON）的路径名。当应用程序尝试将路径名作为常规文件处理时，这通常会导致拒绝服务或信息泄露。

### Description:

The software constructs pathnames from user input, but it does not handle or incorrectly handles a pathname containing a Windows device name such as AUX or CON. This typically leads to denial of service or an information exposure when the application attempts to process the pathname as a regular file.

### 详细描述:

未正确处理虚拟文件名（例如AUX，CON，PRN，COM1，LPT1）可能导致不同类型的漏洞。在某些情况下，攻击者可以通过在URL中注入虚拟文件名来请求设备，这可能导致导致拒绝服务的错误或显示敏感信息的错误页面。允许设备名称绕过过滤的软件系统会冒着攻击者在具有设备名称的文件中注入恶意代码的风险。

### Extended Description:

Not properly handling virtual filenames (e.g. AUX, CON, PRN, COM1, LPT1) can result in different types of vulnerabilities. In some cases an attacker can request a device via injection of a virtual filename in a URL, which may cause an error that leads to a denial of service or an error page that reveals sensitive information. A software system that allows device names to bypass filtering runs the risk of an attacker injecting malicious code in a file with the name of a device.

### 问题背景 (Background Detail):

从历史上看，Windows操作系统中存在一个导致蓝屏死机的错误。即使在该问题被修复之后，DOS设备名称仍然是一个因素。

### 笔记 (Notes):

没有笔记

Weakness ID: 69

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Handling of Windows ::DATA Alternate Data Stream zh: -->Windows :: DATA备用数据流处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法正确阻止对备用数据流（ADS）的访问或检测其使用。

### Description:

The software does not properly prevent access to, or detect usage of, alternate data streams (ADS).

### 详细描述:

攻击者可以使用ADS从系统或文件浏览器工具（如Windows资源管理器）和命令行实用程序中的“dir”隐藏有关文件的信息（例如，大小，进程名称）。或者，攻击者可能能够绕过关联数据分叉的预期访问限制。

### Extended Description:

An attacker can use an ADS to hide information about a file (e.g. size, the name of the process) from a system or file browser tools such as Windows Explorer and 'dir' at the command line utility. Alternately, the attacker might be able to bypass intended access restrictions for the associated data fork.

### 问题背景 (Background Detail):

备用数据流（ADS）首先在Windows NT操作系统中实现，以提供NTFS和Macintosh分层文件系统（HFS）之间的兼容性。在HFS中，数据和资源分叉用于存储有关文件的信息。数据分叉提供有关文件内容的信息，而资源分叉存储元数据（如文件类型）。

### 笔记 (Notes):

存在这种和类似的问题是因为相同的资源可以具有多个标识符，这些标识符指示可以对资源执行哪些行为。

Weakness ID: 71

提交日期 2017-11-08---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> DEPRECATED: Apple '.DS\_Store' zh: -->弃用：Apple'.DS\_Store'*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此条目已被弃用，因为它表示UNIX Hard Link弱点类型的特定观察示例，而不是其自身的个别弱点类型。请参阅CWE-62。

### Description:

This entry has been deprecated as it represents a specific observed example of a UNIX Hard Link weakness type rather than its own individual weakness type. Please refer to CWE-62.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 72

提交日期 2008-11-24---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Apple HFS+ Alternate Data Stream Path zh: -->Apple HFS +备用数据流路径处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法正确处理可能识别HFS +文件系统上文件的数据或资源分支的特殊路径。

### Description:

The software does not properly handle special paths that may identify the data or resource fork of a file on the HFS+ file system.

### 详细描述:

如果软件根据文件名选择要采取的操作，则如果攻击者提供数据或资源分支，则软件可能会采取意外操作。此外，如果软件打算限制对文件的访问，则攻击者仍可能通过请求该文件的数据或资源分支来绕过预期的访问限制。

### Extended Description:

If the software chooses actions to take based on the file name, then if an attacker provides the data or resource fork, the software may take unexpected actions. Further, if the software intends to restrict access to a file, then an attacker might still be able to bypass intended access restrictions by requesting the data or resource fork for that file.

### 问题背景 (Background Detail):

Apple HFS +文件系统允许文件具有多个数据输入流，可通过特殊路径访问。 Mac OS X操作系统提供了一种通过特殊路径和扩展属性访问不同数据输入流的方法： - 资源分支：文件/ .. namedfork / rsrc，文件/ rsrc（不建议使用），xattr：com.apple。 ResourceFork - 数据分支：文件/ .. namedfork / data（仅限Mac OS X v10.5之前的版本）此外，在缺少对多个流的本机支持的文件系统上，资源分叉和文件元数据可以存储在“ .\_“以此为名。也可以通过非便携式API访问Fork。 Forks继承其所属文件的文件系统访问控制。如果文件系统对象的处理依赖于其路径的结构，则程序需要控制对这些路径的访问。

### 笔记 (Notes):

存在这种和类似的问题是因为相同的资源可以具有多个标识符，这些标识符指示可以对资源执行哪些行为。

Weakness ID: 73

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> External Control of File Name or Path zh: -->文件名或路径的外部控制*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件允许用户输入来控制或影响文件系统操作中使用的路径或文件名。

### Description:

The software allows user input to control or influence paths or file names that are used in filesystem operations.

### 详细描述:

这可能允许攻击者访问或修改对应用程序至关重要的系统文件或其他文件。  
满足以下两个条件时会发生路径操作错误：  
  
1.攻击者可以指定文件系统上的操作中使用的路径。  
2.通过指定资源，攻击者获得了不允许的功能。  
  
例如，程序可能使攻击者能够覆盖指定的文件或使用受攻击者控制的配置运行。

### Extended Description:

This could allow an attacker to access or modify system files or other files that are critical to the application.  
Path manipulation errors occur when the following two conditions are met:  
  
1. An attacker can specify a path used in an operation on the filesystem.  
2. By specifying the resource, the attacker gains a capability that would not otherwise be permitted.  
  
For example, the program may give the attacker the ability to overwrite the specified file or run with a configuration controlled by the attacker.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

文件名的外部控制可以是与其他与文件相关的弱点的链中的主要链接，如CanPrecede关系中所示。这是因为软件系统将文件用于许多不同的目的：执行程序，加载代码库，存储应用程序数据，存储配置设置，记录临时数据，充当信号或信号量到其他进程等等。但是，这些弱点确实存在并不总是需要外部控制。例如，链接跟踪弱点（CWE-59）通常涉及攻击者根本无法控制的路径名。外部控制可能是其他问题的结果。例如，在PHP应用程序中，register\_globals设置可以允许攻击者修改程序员认为不可变的变量，启用文件包含（CWE-98）和路径遍历（CWE-22）。使用过多权限（CWE-250）操作可能允许攻击者指定攻击者无法直接读取但可被特权程序访问的输入文件名。缓冲区溢出（CWE-119）可能使攻击者控制与路径名相关的附近内存位置，但攻击者无法直接修改。

Weakness ID: 74

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection') zh: -->下游组件使用的输出中特殊元素的中和不当（'注入'）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用来自上游组件的外部影响输入构造命令，数据结构或记录的全部或部分，但它不会中和或不正确地中和可能修改它被发送到下游组件。

### Description:

The software constructs all or part of a command, data structure, or record using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify how it is parsed or interpreted when it is sent to a downstream component.

### 详细描述:

软件对数据和控制的构成分别有一定的假设。缺乏对用户控制输入的这些假设的验证导致注入问题。注射问题包含各种各样的问题 - 所有问题都以非常不同的方式得到缓解，并且通常是为了改变过程的控制流程。因此，讨论这些弱点的最有效方法是注意将它们归类为注入弱点的独特特征。需要注意的最重要的问题是所有注入问题都有一个共同点 - 即，它们允许将控制平面数据注入用户控制的数据平面。这意味着可以通过合法数据通道发送代码来改变进程的执行，而不使用其他机制。虽然缓冲区溢出和许多其他缺陷涉及使用一些进一步的问题来获得执行，但注入问题仅需要解析数据。这类弱点的最经典的实例是SQL注入和格式字符串漏洞。

### Extended Description:

Software has certain assumptions about what constitutes data and control respectively. It is the lack of verification of these assumptions for user-controlled input that leads to injection problems. Injection problems encompass a wide variety of issues -- all mitigated in very different ways and usually attempted in order to alter the control flow of the process. For this reason, the most effective way to discuss these weaknesses is to note the distinct features which classify them as injection weaknesses. The most important issue to note is that all injection problems share one thing in common -- i.e., they allow for the injection of control plane data into the user-controlled data plane. This means that the execution of the process may be altered by sending code in through legitimate data channels, using no other mechanism. While buffer overflows, and many other flaws, involve the use of some further issue to gain execution, injection problems need only for the data to be parsed. The most classic instantiations of this category of weakness are SQL injection and format string vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在开发视图（CWE-699）中，这被归类为输入验证问题（CWE-20），因为许多人不区分后果/攻击（注入）和防止攻击成功的保护机制。然而，在研究视图（CWE-1000）中，输入验证只是一种潜在的保护机制（输出编码是另一种），并且不正确的输入验证与对其他组件的消息结构的不正确执行之间存在链接关系。与输入验证无直接关系的其他问题（例如竞争条件）可能同样会影响消息结构。

Weakness ID: 75

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Failure to Sanitize Special Elements into a Different Plane (Special Element Injection) zh: -->未将特殊元素消毒到不同的平面（特殊元素注入）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

对于具有控制意义的特殊元素，该软件不能充分过滤用户控制的输入。

### Description:

The software does not adequately filter user-controlled input for special elements with control implications.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 76

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Equivalent Special Elements zh: -->等效特殊元素的中和不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件适当地中和了某些特殊元素，但它不正确地中和了等效的特殊元素。

### Description:

The software properly neutralizes certain special elements, but it improperly neutralizes equivalent special elements.

### 详细描述:

该软件可能具有其认为完整的特殊字符的固定列表。但是，可能存在具有相同含义的替代编码或表示。例如，软件可以过滤掉前导斜杠（/）以防止绝对路径名称，但不会考虑后跟用户名的波浪号（〜），在某些\* nix系统上可以将其扩展为绝对路径名。或者，软件可能在调用外部程序时过滤危险的“-e”命令行开关，但它可能不会考虑“--exec”或具有相同语义的其他开关。

### Extended Description:

The software may have a fixed list of special characters it believes is complete. However, there may be alternate encodings, or representations that also have the same meaning. For example, the software may filter out a leading slash (/) to prevent absolute path names, but does not account for a tilde (~) followed by a user name, which on some \*nix systems could be expanded to an absolute pathname. Alternately, the software might filter a dangerous "-e" command-line switch when calling an external program, but it might not account for "--exec" or other switches that have the same semantics.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 77

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Neutralization of Special Elements used in a Command ('Command Injection') zh: -->命令中使用的特殊元素的不正确中和（'命令注入'）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用来自上游组件的外部影响输入构造全部或部分命令，但它不会中和或不正确地中和可能在将命令发送到下游组件时修改预期命令的特殊元素。

### Description:

The software constructs all or part of a command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended command when it is sent to a downstream component.

### 详细描述:

命令注入漏洞通常在以下情况下发生：  
  
1.数据从不受信任的来源进入应用程序。  
2.数据是应用程序作为命令执行的字符串的一部分。  
3.通过执行该命令，应用程序为攻击者提供了攻击者不具备的特权或能力。  
  
命令注入是包装器程序的常见问题。

### Extended Description:

Command injection vulnerabilities typically occur when:  
  
1. Data enters the application from an untrusted source.  
2. The data is part of a string that is executed as a command by the application.  
3. By executing the command, the application gives an attacker a privilege or capability that the attacker would not otherwise have.  
  
Command injection is a common problem with wrapper programs.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

“命令注入”短语对不同的人具有不同的含义。对于某些人来说，它指的是任何类型的攻击都可以允许攻击者执行他们自己选择的命令，而不管这些命令是如何插入的。因此，命令注入可能来自另一个弱点。该用法还包括这样的情况，其中功能允许用户指定整个命令，然后执行该命令;在CWE中，这种情况可能更好地被视为授权问题（因为攻击者不应该能够指定任意命令。）另一种常见用法，包括CWE-77及其后代，涉及攻击者将分隔符注入到其中的情况。正在构建的命令。

Weakness ID: 78

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') zh: -->OS命令中使用的特殊元素的中和不正确（'OS命令注入'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用来自上游组件的外部影响输入构造全部或部分OS命令，但它不会中和或不正确地中和可能在将其发送到下游组件时修改预期OS命令的特殊元素。

### Description:

The software constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component.

### 详细描述:

这可能允许攻击者直接在操作系统上执行意外的危险命令。这种弱点可能导致攻击者无法直接访问操作系统的环境中存在漏洞，例如在Web应用程序中。或者，如果弱点发生在特权程序中，则可能允许攻击者指定通常无法访问的命令，或者使用攻击者没有的特权来调用备用命令。如果受损的进程不遵循最小权限原则，则问题会更加严重，因为攻击者控制的命令可能会以特殊的系统权限运行，从而增加损害的数量。  
OS命令注入至少有两种子类型：  
  
  
该应用程序打算执行一个由其自己控制的单个固定程序。它打算使用外部提供的输入作为该程序的参数。例如，程序可能使用系统（“nslookup [HOSTNAME]”）来运行nslookup并允许用户提供HOSTNAME，该HOSTNAME用作参数。攻击者无法阻止nslookup执行。但是，如果程序没有从HOSTNAME参数中删除命令分隔符，则攻击者可以将分隔符放入参数中，这允许它们在nslookup完成执行后执行自己的程序。  
应用程序接受一个输入，用于完全选择要运行的程序以及要使用的命令。应用程序只是将整个命令重定向到操作系统。例如，程序可能使用“exec（[COMMAND]）”来执行用户提供的[COMMAND]。如果COMMAND受攻击者控制，则攻击者可以执行任意命令或程序。如果使用exec（）和CreateProcess（）等函数执行命令，则攻击者可能无法在同一行中将多个命令组合在一起。  
  
  
从弱点的角度来看，这些变体代表了不同的程序员错误。在第一个变体中，程序员明确地希望来自不可信方的输入将成为要执行的命令中的参数的一部分。在第二个变体中，程序员不打算让任何不受信任的方可以访问该命令，但程序员可能没有考虑恶意攻击者可以提供输入的替代方式。

### Extended Description:

This could allow attackers to execute unexpected, dangerous commands directly on the operating system. This weakness can lead to a vulnerability in environments in which the attacker does not have direct access to the operating system, such as in web applications. Alternately, if the weakness occurs in a privileged program, it could allow the attacker to specify commands that normally would not be accessible, or to call alternate commands with privileges that the attacker does not have. The problem is exacerbated if the compromised process does not follow the principle of least privilege, because the attacker-controlled commands may run with special system privileges that increases the amount of damage.  
There are at least two subtypes of OS command injection:  
  
  
The application intends to execute a single, fixed program that is under its own control. It intends to use externally-supplied inputs as arguments to that program. For example, the program might use system("nslookup [HOSTNAME]") to run nslookup and allow the user to supply a HOSTNAME, which is used as an argument. Attackers cannot prevent nslookup from executing. However, if the program does not remove command separators from the HOSTNAME argument, attackers could place the separators into the arguments, which allows them to execute their own program after nslookup has finished executing.  
The application accepts an input that it uses to fully select which program to run, as well as which commands to use. The application simply redirects this entire command to the operating system. For example, the program might use "exec([COMMAND])" to execute the [COMMAND] that was supplied by the user. If the COMMAND is under attacker control, then the attacker can execute arbitrary commands or programs. If the command is being executed using functions like exec() and CreateProcess(), the attacker might not be able to combine multiple commands together in the same line.  
  
  
From a weakness standpoint, these variants represent distinct programmer errors. In the first variant, the programmer clearly intends that input from untrusted parties will be part of the arguments in the command to be executed. In the second variant, the programmer does not intend for the command to be accessible to any untrusted party, but the programmer probably has not accounted for alternate ways in which malicious attackers can provide input.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

“OS命令注入”短语对不同的人具有不同的含义。对于某些人来说，它仅指的是攻击者将命令分隔符注入正在调用的应用程序控制程序的参数的情况。对于某些人来说，它指的是任何类型的攻击都可以允许攻击者执行他们自己选择的OS命令。此用法可能包括不受信任的搜索路径弱点（CWE-426），导致应用程序查找并执行攻击者控制的程序。进一步使问题复杂化的是参数注入（CWE-88）允许将备用命令行开关或选项插入命令行，例如“-exec”开关，其目的可能是将后续参数作为命令（例如，这个-exec开关存在于UNIX“find”命令中）。然而，在后一种情况下，CWE-88可被视为CWE-78链中的主要弱点。

Weakness ID: 79

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') zh: -->网页生成期间输入的中和不正确（“跨站点脚本”）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Usable*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在将用户可控输入放入用作提供给其他用户的网页的输出之前，该软件不会中和或不正确地中和用户可控输入。

### Description:

The software does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users.

### 详细描述:

跨站点脚本（XSS）漏洞发生在：  
  
1.不受信任的数据通常来自Web请求进入Web应用程序。  
2. Web应用程序动态生成包含此不受信任数据的网页。  
3.在页面生成期间，应用程序不会阻止数据包含Web浏览器可执行的内容，例如JavaScript，HTML标记，HTML属性，鼠标事件，Flash，ActiveX等。  
4.受害者通过Web浏览器访问生成的Web页面，该Web浏览器包含使用不受信任的数据注入的恶意脚本。  
5.由于脚本来自Web服务器发送的网页，因此受害者的Web浏览器会在Web服务器的域中执行恶意脚本。  
6.这实际上违反了Web浏览器的同源策略的意图，该策略指出一个域中的脚本不应该能够访问资源或在不同的域中运行代码。  
  
XSS有三种主要类型：  
  
  
类型1：反射的XSS（或非持久性） -   
         服务器直接从HTTP请求读取数据并将其反映在HTTP响应中。当攻击者导致受害者向易受攻击的Web应用程序提供危险内容时，会发生反射的XSS攻击，然后将其反射回受害者并由Web浏览器执行。传递恶意内容的最常见机制是将其作为参数包含在公开发布或直接通过电子邮件发送给受害者的URL中。以这种方式构建的URL构成了许多网络钓鱼方案的核心，攻击者诱使受害者访问引用易受攻击网站的URL。在站点将攻击者的内容反映回受害者之后，内容由受害者的浏览器执行。  
  
类型2：存储的XSS（或持久） -   
               应用程序将危险数据存储在数据库，消息论坛，访问者日志或其他可信数据存储中。稍后，危险数据随后被读回应用程序并包含在动态内容中。从攻击者的角度来看，注入恶意内容的最佳位置是在向许多用户或特别有趣的用户显示的区域中。有趣的用户通常在应用程序中具有提升的权限，或者与对攻击者有价值的敏感数据进行交互。如果这些用户之一执行恶意内容，则攻击者可能能够代表用户执行特权操作或访问属于该用户的敏感数据。例如，攻击者可能会将XSS注入日志消息，当管理员查看日志时，可能无法正确处理。  
              
  
类型0：基于DOM的XSS -   
               在基于DOM的XSS中，客户端执行将XSS注入页面;在其他类型中，服务器执行注入。基于DOM的XSS通常涉及发送到客户端的服务器控制的可信脚本，例如在用户提交表单之前对表单执行完整性检查的Javascript。如果服务器提供的脚本处理用户提供的数据，然后将其注入网页（例如使用动态HTML），则可以使用基于DOM的XSS。  
              
  
一旦注入恶意脚本，攻击者就可以执行各种恶意活动。攻击者可以将受害者机器上的私人信息（例如可能包含会话信息的cookie）传输给攻击者。攻击者可以代表受害者向网站发送恶意请求，如果受害者具有管理该网站的管理员权限，则可能对该网站特别危险。网络钓鱼攻击可用于模拟受信任的网站，并欺骗受害者输入密码，允许攻击者破坏受害者在该网站上的帐户。最后，该脚本可以利用Web浏览器本身的漏洞，可能接管受害者的计算机，有时也称为“偷袭黑客”。  
在许多情况下，攻击可以在受害者甚至没有意识到的情况下发起。即使有细心的用户，攻击者也经常使用各种方法来编码攻击的恶意部分，例如URL编码或Unicode，因此请求看起来不那么可疑。

### Extended Description:

Cross-site scripting (XSS) vulnerabilities occur when:  
  
1. Untrusted data enters a web application, typically from a web request.  
2. The web application dynamically generates a web page that contains this untrusted data.  
3. During page generation, the application does not prevent the data from containing content that is executable by a web browser, such as JavaScript, HTML tags, HTML attributes, mouse events, Flash, ActiveX, etc.  
4. A victim visits the generated web page through a web browser, which contains malicious script that was injected using the untrusted data.  
5. Since the script comes from a web page that was sent by the web server, the victim's web browser executes the malicious script in the context of the web server's domain.  
6. This effectively violates the intention of the web browser's same-origin policy, which states that scripts in one domain should not be able to access resources or run code in a different domain.  
  
There are three main kinds of XSS:  
  
  
Type 1: Reflected XSS (or Non-Persistent) -   
 The server reads data directly from the HTTP request and reflects it back in the HTTP response. Reflected XSS exploits occur when an attacker causes a victim to supply dangerous content to a vulnerable web application, which is then reflected back to the victim and executed by the web browser. The most common mechanism for delivering malicious content is to include it as a parameter in a URL that is posted publicly or e-mailed directly to the victim. URLs constructed in this manner constitute the core of many phishing schemes, whereby an attacker convinces a victim to visit a URL that refers to a vulnerable site. After the site reflects the attacker's content back to the victim, the content is executed by the victim's browser.  
  
Type 2: Stored XSS (or Persistent) -   
 The application stores dangerous data in a database, message forum, visitor log, or other trusted data store. At a later time, the dangerous data is subsequently read back into the application and included in dynamic content. From an attacker's perspective, the optimal place to inject malicious content is in an area that is displayed to either many users or particularly interesting users. Interesting users typically have elevated privileges in the application or interact with sensitive data that is valuable to the attacker. If one of these users executes malicious content, the attacker may be able to perform privileged operations on behalf of the user or gain access to sensitive data belonging to the user. For example, the attacker might inject XSS into a log message, which might not be handled properly when an administrator views the logs.  
   
  
Type 0: DOM-Based XSS -   
 In DOM-based XSS, the client performs the injection of XSS into the page; in the other types, the server performs the injection. DOM-based XSS generally involves server-controlled, trusted script that is sent to the client, such as Javascript that performs sanity checks on a form before the user submits it. If the server-supplied script processes user-supplied data and then injects it back into the web page (such as with dynamic HTML), then DOM-based XSS is possible.  
   
  
Once the malicious script is injected, the attacker can perform a variety of malicious activities. The attacker could transfer private information, such as cookies that may include session information, from the victim's machine to the attacker. The attacker could send malicious requests to a web site on behalf of the victim, which could be especially dangerous to the site if the victim has administrator privileges to manage that site. Phishing attacks could be used to emulate trusted web sites and trick the victim into entering a password, allowing the attacker to compromise the victim's account on that web site. Finally, the script could exploit a vulnerability in the web browser itself possibly taking over the victim's machine, sometimes referred to as "drive-by hacking."  
In many cases, the attack can be launched without the victim even being aware of it. Even with careful users, attackers frequently use a variety of methods to encode the malicious portion of the attack, such as URL encoding or Unicode, so the request looks less suspicious.

### 问题背景 (Background Detail):

同源策略相同的源策略规定浏览器应将在给定网站或“源”上运行的脚本可访问的资源限制为与客户端上与该网站关联的资源，而不是客户端资源任何其他网站或“起源”。目标是防止一个站点修改或读取不相关站点的内容。由于万维网涉及许多站点之间的交互，因此该策略对于浏览器的实施非常重要。域引用XSS时网站的域大致相当于连接客户端上与该网站关联的资源。也就是说，域可以被认为是浏览器为用户与该特定站点的交互存储的所有资源。

### 笔记 (Notes):

XSS和CSRF（CWE-352）之间可能存在密切关系。攻击者可能使用CSRF来诱骗受害者向请求包含XSS有效负载的服务器提交请求。一个众所周知的例子就是MySpace上的Samy蠕虫[REF-956]。蠕虫使用XSS将恶意HTML序列插入用户的个人资料中，并将攻击者添加为MySpace朋友。然后，该受害者的MySpace朋友将执行有效负载以修改他们自己的配置文件，从而导致蠕虫以指数方式传播。由于受害者本身并未故意插入恶意脚本，因此CSRF是一个根本原因。

Weakness ID: 80

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS) zh: -->网页中与脚本相关的HTML标记的中和不当（基本XSS）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和特殊字符，例如“<”，“>”和“＆”，当它们被发送到下游组件时可被解释为Web脚本元素处理网页。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special characters such as "<", ">", and "&" that could be interpreted as web-scripting elements when they are sent to a downstream component that processes web pages.

### 详细描述:

这可以允许将这些字符视为控制字符，其在用户会话的上下文中在客户端执行。虽然这可以归类为注入问题，但更相关的问题是在将这些特殊字符显示给用户之前将这些特殊字符不正确地转换到相应的上下文相关实体。

### Extended Description:

This may allow such characters to be treated as control characters, which are executed client-side in the context of the user's session. Although this can be classified as an injection problem, the more pertinent issue is the improper conversion of such special characters to respective context-appropriate entities before displaying them to the user.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 81

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Script in an Error Message Web Page zh: -->错误消息网页中脚本的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和可能在发送到错误页面时被解释为Web脚本元素的特殊字符。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special characters that could be interpreted as web-scripting elements when they are sent to an error page.

### 详细描述:

错误页面可能包括自定义的403 Forbidden或404 Not Found页面。  
当攻击者可以触发包含未中和输入的错误时，可能会发生跨站点脚本攻击。

### Extended Description:

Error pages may include customized 403 Forbidden or 404 Not Found pages.  
When an attacker can trigger an error that contains unneutralized input, then cross-site scripting attacks may be possible.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 82

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Script in Attributes of IMG Tags in a Web Page zh: -->网页中IMG标记属性中脚本的中和不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序不会中和或错误地中和HTML IMG标记属性中的脚本元素，例如src属性。

### Description:

The web application does not neutralize or incorrectly neutralizes scripting elements within attributes of HTML IMG tags, such as the src attribute.

### 详细描述:

攻击者可以将XSS攻击嵌入到流式传输然后在受害者浏览器中执行的IMG属性（例如SRC）的值中。请注意，当页面加载到用户的浏览器中时，漏洞会自动执行。

### Extended Description:

Attackers can embed XSS exploits into the values for IMG attributes (e.g. SRC) that is streamed and then executed in a victim's browser. Note that when the page is loaded into a user's browsers, the exploit will automatically execute.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 83

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Script in Attributes in a Web Page zh: -->网页属性中脚本的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会中和或错误地中和“javascript：”或来自标签内危险属性的其他URI，例如onmouseover，onload，onerror或style。

### Description:

The software does not neutralize or incorrectly neutralizes "javascript:" or other URIs from dangerous attributes within tags, such as onmouseover, onload, onerror, or style.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 84

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Encoded URI Schemes in a Web Page zh: -->网页中编码URI方案的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序不正确地中和了用URI编码伪装的可执行脚本的用户控制输入。

### Description:

The web application improperly neutralizes user-controlled input for executable script disguised with URI encodings.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 85

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Doubled Character XSS Manipulations zh: -->双字符XSS操作*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序不会过滤用户控制的输入，因为可执行脚本使用相关字符加倍来伪装。

### Description:

The web application does not filter user-controlled input for executable script disguised using doubling of the involved characters.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 86

提交日期 2008-09-09---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Invalid Characters in Identifiers in Web Pages zh: -->网页标识符中无效字符的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会中和或错误地中和标记名称，URI方案和其他标识符中间的无效字符或字节序列。

### Description:

The software does not neutralize or incorrectly neutralizes invalid characters or byte sequences in the middle of tag names, URI schemes, and other identifiers.

### 详细描述:

某些Web浏览器可能会删除这些序列，从而导致输出可能具有意外的控制含义。例如，软件可能会尝试删除“javascript：”URI方案，但“java％00script：”URI可能会绕过此检查，并且仍会被某些浏览器呈现为活动javascript，从而允许XSS或其他攻击。

### Extended Description:

Some web browsers may remove these sequences, resulting in output that may have unintended control implications. For example, the software may attempt to remove a "javascript:" URI scheme, but a "java%00script:" URI may bypass this check and still be rendered as active javascript by some browsers, allowing XSS or other attacks.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 87

提交日期 2008-09-09---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Alternate XSS Syntax zh: -->备用XSS语法的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会中和或错误地中和用户控制的输入以获得备用脚本语法。

### Description:

The software does not neutralize or incorrectly neutralizes user-controlled input for alternate script syntax.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 88

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Argument Injection or Modification zh: -->参数注入或修改*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件没有充分划分传递给另一个控件领域中的组件的参数，允许提供备用参数，从而导致潜在的安全相关更改。

### Description:

The software does not sufficiently delimit the arguments being passed to a component in another control sphere, allowing alternate arguments to be provided, leading to potentially security-relevant changes.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在一个抽象层，这可以与具有空白问题的其他弱点重叠，例如，将javascript注入HTML标记的属性。

Weakness ID: 89

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') zh: -->SQL命令中使用的特殊元素的中和不正确（'SQL注入'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用来自上游组件的外部影响输入构造全部或部分SQL命令，但它不会中和或不正确地中和可能在发送到下游组件时修改预期SQL命令的特殊元素。

### Description:

The software constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component.

### 详细描述:

如果没有在用户可控输入中充分删除或引用SQL语法，则生成的SQL查询可以将这些输入解释为SQL而不是普通用户数据。这可用于更改查询逻辑以绕过安全检查，或插入修改后端数据库的其他语句，可能包括执行系统命令。  
SQL注入已经成为数据库驱动的网站的常见问题。该漏洞很容易被检测到，并且容易被利用，因此，任何具有最小用户群的站点或软件包都可能遭受此类攻击。这个缺陷取决于SQL在控制平面和数据平面之间没有真正区别的事实。

### Extended Description:

Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the back-end database, possibly including execution of system commands.  
SQL injection has become a common issue with database-driven web sites. The flaw is easily detected, and easily exploited, and as such, any site or software package with even a minimal user base is likely to be subject to an attempted attack of this kind. This flaw depends on the fact that SQL makes no real distinction between the control and data planes.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

SQL注入可能是由特殊字符管理不善，MAID或黑名单/白名单问题引起的。它可能是身份验证错误的主要原因。

Weakness ID: 90

提交日期 2008-04-11---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Neutralization of Special Elements used in an LDAP Query ('LDAP Injection') zh: -->LDAP查询中使用的特殊元素的中和不正确（'LDAP注入'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用来自上游组件的外部影响输入构造全部或部分LDAP查询，但它不会中和或不正确地中和可能在将其发送到下游组件时修改预期LDAP查询的特殊元素。

### Description:

The software constructs all or part of an LDAP query using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended LDAP query when it is sent to a downstream component.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

因素：导致特殊字符管理不善，MAID或黑名单/白名单问题。可以是主要的身份验证和验证错误。

Weakness ID: 91

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> XML Injection (aka Blind XPath Injection) zh: -->XML注入（又称盲注XPath注入）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法正确中和XML中使用的特殊元素，允许攻击者在终端系统处理之前修改XML的语法，内容或命令。

### Description:

The software does not properly neutralize special elements that are used in XML, allowing attackers to modify the syntax, content, or commands of the XML before it is processed by an end system.

### 详细描述:

在XML中，特殊元素可以包括保留字或字符，例如“<”，“>”，“”和“＆”，然后可以用于添加新数据或修改XML语法。

### Extended Description:

Within XML, special elements could include reserved words or characters such as "<", ">", """, and "&", which could then be used to add new data or modify XML syntax.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目的描述通常适用于XML，但名称包括“盲XPath注入”，它与CWE-643关系更密切。因此，可能需要弃用此条目或将其转换为常规类别 - 尽管CWE-643或CWE-652不涵盖对原始XML的注入。

Weakness ID: 92

提交日期 2008-10-14---> 修改日期 2010-06-21

* **Weakness Name:** *en: --> DEPRECATED: Improper Sanitization of Custom Special Characters zh: -->弃用：自定义特殊字符的不当消毒*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此条目已被弃用。它最初来自PLOVER，它有时定义“其他”和“杂项”类别，以满足分类法的详尽性要求。在CWE的上下文中，在映射情况下优选使用更抽象的条目。 CWE-75是更合适的映射。

### Description:

This entry has been deprecated. It originally came from PLOVER, which sometimes defined "other" and "miscellaneous" categories in order to satisfy exhaustiveness requirements for taxonomies. Within the context of CWE, the use of a more abstract entry is preferred in mapping situations. CWE-75 is a more appropriate mapping.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 93

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of CRLF Sequences ('CRLF Injection') zh: -->CRLF序列的中和不当（'CRLF注射'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用CRLF（回车换行）作为特殊元素，例如，分隔行或记录，但它不会中和或错误地中和输入中的CRLF序列。

### Description:

The software uses CRLF (carriage return line feeds) as a special element, e.g. to separate lines or records, but it does not neutralize or incorrectly neutralizes CRLF sequences from inputs.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可能研究不足，但由于对HTTP响应分裂的兴趣，2005年更加突出。

Weakness ID: 94

提交日期 2009-01-12---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Control of Generation of Code ('Code Injection') zh: -->代码生成控制不当（'代码注入'）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用来自上游组件的外部影响输入构造全部或部分代码段，但它不会中和或不正确地中和可能修改预期代码段的语法或行为的特殊元素。

### Description:

The software constructs all or part of a code segment using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the syntax or behavior of the intended code segment.

### 详细描述:

当软件允许用户的输入包含代码语法时，攻击者可能以这样的方式制作代码，使其改变软件的预期控制流程。这种改变可能导致任意代码执行。  
注射问题包含各种各样的问题 - 所有问题都以非常不同的方式得到缓解。因此，讨论这些弱点的最有效方法是注意将它们归类为注入弱点的独特特征。需要注意的最重要的问题是所有注入问题都有一个共同点 - 即，它们允许将控制平面数据注入用户控制的数据平面。这意味着可以通过合法数据通道发送代码来改变进程的执行，而不使用其他机制。虽然缓冲区溢出和许多其他缺陷涉及使用一些进一步的问题来获得执行，但注入问题仅需要解析数据。这类弱点的最经典的实例是SQL注入和格式字符串漏洞。

### Extended Description:

When software allows a user's input to contain code syntax, it might be possible for an attacker to craft the code in such a way that it will alter the intended control flow of the software. Such an alteration could lead to arbitrary code execution.  
Injection problems encompass a wide variety of issues -- all mitigated in very different ways. For this reason, the most effective way to discuss these weaknesses is to note the distinct features which classify them as injection weaknesses. The most important issue to note is that all injection problems share one thing in common -- i.e., they allow for the injection of control plane data into the user-controlled data plane. This means that the execution of the process may be altered by sending code in through legitimate data channels, using no other mechanism. While buffer overflows, and many other flaws, involve the use of some further issue to gain execution, injection problems need only for the data to be parsed. The most classic instantiations of this category of weakness are SQL injection and format string vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

许多这些弱点未得到充分研究和研究不足，术语不够精确。

Weakness ID: 95

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection') zh: -->动态评估代码中指令的不正确中和（'Eval Injection'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件接收来自上游组件的输入，但在动态评估调用（例如“eval”）中使用输入之前，它不会中和或不正确地中和代码语法。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes code syntax before using the input in a dynamic evaluation call (e.g. "eval").

### 详细描述:

这可能允许攻击者执行任意代码，或者至少修改可以执行的代码。

### Extended Description:

This may allow an attacker to execute arbitrary code, or at least modify what code can be executed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

因素：特殊字符错误可以在增加可注入的代码种类方面发挥作用，尽管某些漏洞根本不需要特殊字符，例如：当可以引用没有参数的单个函数并且不需要终止符时。

Weakness ID: 96

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Directives in Statically Saved Code ('Static Code Injection') zh: -->静态保存代码中指令的不正确中和（'静态代码注入'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但在将输入插入可执行资源（例如库，配置文件或模板）之前，它不会中和或不正确地中和代码语法。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes code syntax before inserting the input into an executable resource, such as a library, configuration file, or template.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

“HTML注入”（参见CWE-79：XSS）可以被认为是一个例子，但是代码是在客户端而不是服务器端注入和执行的。服务器端包含（SSI）是直接静态代码注入的示例。

Weakness ID: 97

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Server-Side Includes (SSI) Within a Web Page zh: -->网页中服务器端包含（SSI）的不正确中和*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件生成一个网页，但不会中和或不正确地中和可被解释为服务器端包含（SSI）指令的用户可控输入。

### Description:

The software generates a web page, but does not neutralize or incorrectly neutralizes user-controllable input that could be interpreted as a server-side include (SSI) directive.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能是XSS / HTML注入的结果，因为可能涉及相同的特殊字符。但是，这是服务器端代码执行，而不是客户端。

Weakness ID: 98

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Control of Filename for Include/Require Statement in PHP Program ('PHP Remote File Inclusion') zh: -->PHP程序中Include / Require语句的文件名控制不当（'PHP远程文件包含'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

PHP应用程序从上游组件接收输入，但在“require”，“include”或类似函数中使用之前，它不会限制或错误地限制输入。

### Description:

The PHP application receives input from an upstream component, but it does not restrict or incorrectly restricts the input before its usage in "require," "include," or similar functions.

### 详细描述:

在PHP的某些版本和配置中，这可以允许攻击者指定远程位置的URL，软件将从该位置获取要执行的代码。在与路径遍历相关联的其他情况下，攻击者可以指定可能包含可由PHP解析的可执行语句的本地文件。

### Extended Description:

In certain versions and configurations of PHP, this can allow an attacker to specify a URL to a remote location from which the software will obtain the code to execute. In other cases in association with path traversal, the attacker can specify a local file that may contain executable statements that can be parsed by PHP.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这通常是其他弱点的功能性后果。它通常是具有其他因素的多因素（例如MAID），尽管并非所有包含错误都涉及假设不可变数据。直接请求弱点经常发挥作用。可以在本地包含问题中重叠目录遍历。

Weakness ID: 99

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Control of Resource Identifiers ('Resource Injection') zh: -->资源标识符控制不当（'资源注入'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件接收来自上游组件的输入，但在将输入用作可能在预期控制范围之外的资源的标识符之前，它不限制或不正确地限制输入。

### Description:

The software receives input from an upstream component, but it does not restrict or incorrectly restricts the input before it is used as an identifier for a resource that may be outside the intended sphere of control.

### 详细描述:

满足以下两个条件时会发生资源注入问题：  
  
  
攻击者可以指定用于访问系统资源的标识符。例如，攻击者可能能够指定要打开的文件的名称的一部分或要使用的端口号。  
通过指定资源，攻击者获得了不允许的功能。例如，程序可能使攻击者能够覆盖指定的文件，使用受攻击者控制的配置运行，或者将敏感信息传输到第三方服务器。  
  
  
这可以使攻击者能够访问或修改受保护的系统资源。

### Extended Description:

A resource injection issue occurs when the following two conditions are met:  
  
  
An attacker can specify the identifier used to access a system resource. For example, an attacker might be able to specify part of the name of a file to be opened or a port number to be used.  
By specifying the resource, the attacker gains a capability that would not otherwise be permitted. For example, the program may give the attacker the ability to overwrite the specified file, run with a configuration controlled by the attacker, or transmit sensitive information to a third-party server.  
  
  
This may enable an attacker to access or modify otherwise protected system resources.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

涉及存储在文件系统上的资源的资源注入通过名称路径操作（CWE-73）进行。

Weakness ID: 102

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Struts: Duplicate Validation Forms zh: -->Struts：重复的验证表单*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序使用多个具有相同名称的验证表单，这可能会导致Struts验证程序验证程序员不期望的表单。

### Description:

The application uses multiple validation forms with the same name, which might cause the Struts Validator to validate a form that the programmer does not expect.

### 详细描述:

如果两个验证表单具有相同的名称，Struts Validator会任意选择其中一个表单用于输入验证并丢弃另一个表单。这个决定可能与程序员的期望不符，可能导致最终的弱点。此外，它表明验证逻辑不是最新的，并且可以指示存在其他更微妙的验证错误。

### Extended Description:

If two validation forms have the same name, the Struts Validator arbitrarily chooses one of the forms to use for input validation and discards the other. This decision might not correspond to the programmer's expectations, possibly leading to resultant weaknesses. Moreover, it indicates that the validation logic is not up-to-date, and can indicate that other, more subtle validation errors are present.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 103

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Struts: Incomplete validate() Method Definition zh: -->Struts：不完整的validate（）方法定义*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序具有验证器表单，该表单既不定义validate（）方法，也不定义validate（）方法但不调用super.validate（）。

### Description:

The application has a validator form that either does not define a validate() method, or defines a validate() method but does not call super.validate().

### 详细描述:

如果不调用super.validate（），则验证框架无法根据验证表单检查表单的内容。换句话说，将针对给定表单禁用验证框架。

### Extended Description:

If you do not call super.validate(), the Validation Framework cannot check the contents of the form against a validation form. In other words, the validation framework will be disabled for the given form.

### 问题背景 (Background Detail):

Struts Validator使用表单的validate（）方法根据相关验证表单中指定的约束检查表单属性的内容。这意味着以下类具有validate（）方法，该方法是验证框架的一部分：ValidatorForm，ValidatorActionForm，DynaValidatorForm和DynaValidatorActionForm。如果您创建了一个扩展其中一个类的类，并且您的类通过重写validate（）方法来实现自定义验证逻辑，则必须在validate（）实现中调用super.validate（）。

### 笔记 (Notes):

这可能会引入与缺少输入验证相关的其他弱点。

Weakness ID: 104

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Struts: Form Bean Does Not Extend Validation Class zh: -->Struts：Form Bean不会扩展验证类*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果表单bean没有扩展Validator框架的ActionForm子类，它可以将应用程序暴露给与输入验证不足相关的其他弱点。

### Description:

If a form bean does not extend an ActionForm subclass of the Validator framework, it can expose the application to other weaknesses related to insufficient input validation.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

为了使用Struts Validator，表单必须扩展以下之一：ValidatorForm，ValidatorActionForm，DynaValidatorActionForm和DynaValidatorForm。您必须扩展其中一个类，因为Struts Validator通过在这些类中实现validate（）方法而与您的应用程序绑定。从ActionForm和DynaActionForm类派生的表单不能使用Struts Validator。

### 笔记 (Notes):

没有笔记

Weakness ID: 105

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Struts: Form Field Without Validator zh: -->Struts：没有验证器的表单字段*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序的表单字段未通过相应的验证表单进行验证，这可能会引入与输入验证不足相关的其他弱点。

### Description:

The application has a form field that is not validated by a corresponding validation form, which can introduce other weaknesses related to insufficient input validation.

### 详细描述:

即使是单个输入字段省略验证也可能为攻击者提供破坏应用程序所需的余地。尽管J2EE应用程序通常不容易受到内存损坏攻击，但如果J2EE应用程序与不执行数组边界检查的本机代码接口，则攻击者可能能够在J2EE应用程序中使用输入验证错误来启动缓冲区溢出攻击。

### Extended Description:

Omitting validation for even a single input field may give attackers the leeway they need to compromise the application. Although J2EE applications are not generally susceptible to memory corruption attacks, if a J2EE application interfaces with native code that does not perform array bounds checking, an attacker may be able to use an input validation mistake in the J2EE application to launch a buffer overflow attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 106

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Struts: Plug-in Framework not in Use zh: -->Struts：未使用的插件框架*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当应用程序不使用输入验证框架（如Struts Validator）时，引入与输入验证不足相关的弱点的风险更大。

### Description:

When an application does not use an input validation framework such as the Struts Validator, there is a greater risk of introducing weaknesses related to insufficient input validation.

### 详细描述:

未经检查的输入是J2EE应用程序中漏洞的主要原因。未经检查的输入会导致跨站点脚本，进程控制和SQL注入漏洞等。  
尽管J2EE应用程序通常不容易受到内存损坏攻击，但如果J2EE应用程序与不执行数组边界检查的本机代码接口，则攻击者可能能够在J2EE应用程序中使用输入验证错误来启动缓冲区溢出攻击。

### Extended Description:

Unchecked input is the leading cause of vulnerabilities in J2EE applications. Unchecked input leads to cross-site scripting, process control, and SQL injection vulnerabilities, among others.  
Although J2EE applications are not generally susceptible to memory corruption attacks, if a J2EE application interfaces with native code that does not perform array bounds checking, an attacker may be able to use an input validation mistake in the J2EE application to launch a buffer overflow attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 107

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Struts: Unused Validation Form zh: -->Struts：未使用的验证表单*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

未使用的验证表单表明验证逻辑不是最新的。

### Description:

An unused validation form indicates that validation logic is not up-to-date.

### 详细描述:

当开发人员删除或重命名动作表单映射时，很容易忘记更新验证逻辑。验证逻辑未得到正确维护的一个迹象是存在未使用的验证表单。

### Extended Description:

It is easy for developers to forget to update validation logic when they remove or rename action form mappings. One indication that validation logic is not being properly maintained is the presence of an unused validation form.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 108

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Struts: Unvalidated Action Form zh: -->Struts：未经验证的行动表格*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

每个行动表格都必须有相应的验证表格。

### Description:

Every Action Form must have a corresponding validation form.

### 详细描述:

如果Struts Action Form Mapping指定了一个表单，它必须在Struts Validator下定义一个验证表单。

### Extended Description:

If a Struts Action Form Mapping specifies a form, it must have a validation form defined under the Struts Validator.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 109

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Struts: Validator Turned Off zh: -->Struts：验证器关闭*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

已关闭通过Struts bean自动筛选，这会禁用Struts Validator和自定义验证逻辑。这使应用程序暴露于与输入验证不足相关的其他弱点。

### Description:

Automatic filtering via a Struts bean has been turned off, which disables the Struts Validator and custom validation logic. This exposes the application to other weaknesses related to insufficient input validation.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

演示示例中的Action Form映射禁用了表单的validate（）方法。 Struts bean：write标签自动编码特殊的HTML字符，替换<with“＆lt;”和a>“＆gt;”。可以通过将filter =“false”指定为标记的属性来禁用此操作，以禁用指定的JSP页面。但是，被禁用会使这些页面容易受到跨站点脚本攻击。攻击者可能能够插入恶意脚本作为用户输入来写入这些JSP页面。

Weakness ID: 110

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Struts: Validator Without Form Field zh: -->Struts：没有表单字段的验证器*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

未出现在与其关联的表单中的验证字段表示验证逻辑已过期。

### Description:

Validation fields that do not appear in forms they are associated with indicate that the validation logic is out of date.

### 详细描述:

开发人员在更改ActionForm类时很容易忘记更新验证逻辑。验证逻辑未得到正确维护的一个迹象是操作表单和验证表单之间的不一致。  
尽管J2EE应用程序通常不容易受到内存损坏攻击，但如果J2EE应用程序与不执行数组边界检查的本机代码接口，则攻击者可能能够在J2EE应用程序中使用输入验证错误来启动缓冲区溢出攻击。

### Extended Description:

It is easy for developers to forget to update validation logic when they make changes to an ActionForm class. One indication that validation logic is not being properly maintained is inconsistencies between the action form and the validation form.  
Although J2EE applications are not generally susceptible to memory corruption attacks, if a J2EE application interfaces with native code that does not perform array bounds checking, an attacker may be able to use an input validation mistake in the J2EE application to launch a buffer overflow attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 111

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Direct Use of Unsafe JNI zh: -->直接使用不安全的JNI*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当Java应用程序使用Java本机接口（JNI）来调用用另一种编程语言编写的代码时，它可以将应用程序暴露给该代码中的弱点，即使这些弱点不能在Java中出现。

### Description:

When a Java application uses the Java Native Interface (JNI) to call code written in another programming language, it can expose the application to weaknesses in that code, even if those weaknesses cannot occur in Java.

### 详细描述:

程序员可能认为理所当然的许多安全功能根本不适用于本机代码，因此您必须仔细检查所有此类代码是否存在潜在问题。用于实现本机代码的语言可能更容易受到缓冲区溢出和其他攻击的影响。本机代码不受运行时环境强制执行的安全功能的保护，例如强类型和数组边界检查。

### Extended Description:

Many safety features that programmers may take for granted simply do not apply for native code, so you must carefully review all such code for potential problems. The languages used to implement native code may be more susceptible to buffer overflows and other attacks. Native code is unprotected by the security features enforced by the runtime environment, such as strong typing and array bounds checking.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 112

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Missing XML Validation zh: -->缺少XML验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接受来自不受信任来源的XML，但不会根据正确的模式验证XML。

### Description:

The software accepts XML from an untrusted source but does not validate the XML against the proper schema.

### 详细描述:

大多数成功的攻击始于违反程序员的假设。通过接受XML文档而不针对DTD或XML模式验证它，程序员为攻击者留下了一扇门，以便提供意外，不合理或恶意的输入。

### Extended Description:

Most successful attacks begin with a violation of the programmer's assumptions. By accepting an XML document without validating it against a DTD or XML schema, the programmer leaves a door open for attackers to provide unexpected, unreasonable, or malicious input.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 113

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of CRLF Sequences in HTTP Headers ('HTTP Response Splitting') zh: -->HTTP标头中CRLF序列的不正确中和（'HTTP响应拆分'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件从上游组件接收数据，但在数据包含在传出HTTP标头中之前，不会中和或错误地中和CR和LF字符。

### Description:

The software receives data from an upstream component, but does not neutralize or incorrectly neutralizes CR and LF characters before the data is included in outgoing HTTP headers.

### 详细描述:

在HTTP标头中包含未经验证的数据允许攻击者指定浏览器呈现的整个HTTP响应。当HTTP请求包含意外CR（回车，也由％0d或\ r \ n）和LF（换行，也由％0a或\ n给出）字符时，服务器可能会响应一个被解释为两个不同的输出流HTTP响应（而不是一个）。攻击者可以控制第二个响应并加载攻击，例如跨站点脚本和缓存中毒攻击。  
在以下情况下可能存在HTTP响应拆分弱点：  
  
  
数据通过不受信任的来源进入Web应用程序，最常见的是HTTP请求。  
数据包含在发送给Web用户的HTTP响应标头中，而不会对恶意字符进行验证。

### Extended Description:

Including unvalidated data in an HTTP header allows an attacker to specify the entirety of the HTTP response rendered by the browser. When an HTTP request contains unexpected CR (carriage return, also given by %0d or \r) and LF (line feed, also given by %0a or \n) characters the server may respond with an output stream that is interpreted as two different HTTP responses (instead of one). An attacker can control the second response and mount attacks such as cross-site scripting and cache poisoning attacks.  
HTTP response splitting weaknesses may be present when:  
  
  
Data enters a web application through an untrusted source, most frequently an HTTP request.  
The data is included in an HTTP response header sent to a web user without being validated for malicious characters.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在使用中介的环境中，HTTP响应拆分可能只是多因素。

Weakness ID: 114

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Process Control zh: -->过程控制*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

从不受信任的源或不受信任的环境执行命令或加载库可能导致应用程序代表攻击者执行恶意命令（和有效负载）。

### Description:

Executing commands or loading libraries from an untrusted source or in an untrusted environment can cause an application to execute malicious commands (and payloads) on behalf of an attacker.

### 详细描述:

进程控制漏洞有两种形式：1。攻击者可以更改程序执行的命令：攻击者明确控制命令的内容。 2.攻击者可以更改命令执行的环境：攻击者隐式控制命令的含义。当数据从不受信任的源进入应用程序并且数据用作表示应用程序执行的命令的字符串的一部分时，会发生第一种类型的进程控制漏洞。通过执行该命令，应用程序为攻击者提供了攻击者无法拥有的特权或能力。

### Extended Description:

Process control vulnerabilities take two forms: 1. An attacker can change the command that the program executes: the attacker explicitly controls what the command is. 2. An attacker can change the environment in which the command executes: the attacker implicitly controls what the command means. Process control vulnerabilities of the first type occur when either data enters the application from an untrusted source and the data is used as part of a string representing a command that is executed by the application. By executing the command, the application gives an attacker a privilege or capability that the attacker would not otherwise have.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 115

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Misinterpretation of Input zh: -->对输入的误解*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件以安全相关的方式错误解释了来自攻击者或其他产品的输入。

### Description:

The software misinterprets an input, whether from an attacker or another product, in a security-relevant fashion.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这个概念需要进一步研究。它可能是几个弱点的一个因素，也可能是一个因素。重叠多重解释错误（MIE）。

Weakness ID: 116

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Encoding or Escaping of Output zh: -->输出的编码或转义不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件准备用于与另一个组件通信的结构化消息，但是数据的编码或转义要么丢失要么不正确。结果，不保留消息的预期结构。

### Description:

The software prepares a structured message for communication with another component, but encoding or escaping of the data is either missing or done incorrectly. As a result, the intended structure of the message is not preserved.

### 详细描述:

不正确的编码或转义可能允许攻击者更改发送到另一个组件的命令，而是插入恶意命令。  
大多数软件遵循某种协议，该协议使用结构化消息进行组件之间的通信，例如查询或命令。这些结构化消息可以包含散布有元数据或控制信息的原始数据。例如，“GET /index.html HTTP / 1.1”是一个结构化消息，其中包含一个带有单个参数（“/index.html”）的命令（“GET”）和有关正在使用哪个协议版本的元数据（“HTTP / 1.1" ）。  
如果应用程序使用攻击者提供的输入来构造结构化消息而没有正确编码或转义，则攻击者可以插入特殊字符，这些特征将导致数据被解释为控制信息或元数据。因此，接收输出的组件将执行错误的操作，或以其他方式错误地解释数据。

### Extended Description:

Improper encoding or escaping can allow attackers to change the commands that are sent to another component, inserting malicious commands instead.  
Most software follows a certain protocol that uses structured messages for communication between components, such as queries or commands. These structured messages can contain raw data interspersed with metadata or control information. For example, "GET /index.html HTTP/1.1" is a structured message containing a command ("GET") with a single argument ("/index.html") and metadata about which protocol version is being used ("HTTP/1.1").  
If an application uses attacker-supplied inputs to construct a structured message without properly encoding or escaping, then the attacker could insert special characters that will cause the data to be interpreted as control information or metadata. Consequently, the component that receives the output will perform the wrong operations, or otherwise interpret the data incorrectly.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这种弱点是与注射相关的所有弱点（CWE-74）的主要原因，因为注射的固有性质涉及违反结构化信息。

Weakness ID: 117

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Output Neutralization for Logs zh: -->日志的输出中和不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会中和或错误地中和写入日志的输出。

### Description:

The software does not neutralize or incorrectly neutralizes output that is written to logs.

### 详细描述:

这可能允许攻击者伪造日志条目或将恶意内容注入日志。  
在以下情况下发生日志伪造漏洞：  
  
  
数据从不受信任的来源进入应用程序。  
数据将写入应用程序或系统日志文件。

### Extended Description:

This can allow an attacker to forge log entries or inject malicious content into logs.  
Log forging vulnerabilities occur when:  
  
  
Data enters an application from an untrusted source.  
The data is written to an application or system log file.

### 问题背景 (Background Detail):

应用程序通常使用日志文件来存储事件或事务的历史记录，以供以后查看，统计信息收集或调试。根据应用程序的性质，审查日志文件的任务可以根据需要手动执行，也可以使用自动剔除重要事件或趋势信息日志的工具自动执行。

### 笔记 (Notes):

没有笔记

Weakness ID: 118

提交日期 2008-09-09---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incorrect Access of Indexable Resource ('Range Error') zh: -->可索引资源的不正确访问（“范围错误”）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不限制或不正确地限制使用索引或指针（例如内存或文件）访问的资源边界内的操作。

### Description:

The software does not restrict or incorrectly restricts operations within the boundaries of a resource that is accessed using an index or pointer, such as memory or files.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 119

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Restriction of Operations within the Bounds of a Memory Buffer zh: -->内存缓冲区内的操作限制不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Usable*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件对内存缓冲区执行操作，但它可以读取或写入缓冲区预期边界之外的内存位置。

### Description:

The software performs operations on a memory buffer, but it can read from or write to a memory location that is outside of the intended boundary of the buffer.

### 详细描述:

某些语言允许直接寻址内存位置，并且不会自动确保这些位置对正在引用的内存缓冲区有效。这可以导致对可能与其他变量，数据结构或内部程序数据相关联的存储器位置执行读或写操作。  
因此，攻击者可能能够执行任意代码，更改预期的控制流，读取敏感信息或导致系统崩溃。

### Extended Description:

Certain languages allow direct addressing of memory locations and do not automatically ensure that these locations are valid for the memory buffer that is being referenced. This can cause read or write operations to be performed on memory locations that may be associated with other variables, data structures, or internal program data.  
As a result, an attacker may be able to execute arbitrary code, alter the intended control flow, read sensitive information, or cause the system to crash.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在没有内存管理支持的任何编程语言中，都可以尝试在内存缓冲区范围之外进行操作，但结果会因语言，平台和芯片架构而有很大差异。

Weakness ID: 120

提交日期 2008-10-14---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Buffer Copy without Checking Size of Input ('Classic Buffer Overflow') zh: -->缓冲区复制而不检查输入大小（'经典缓冲区溢出'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

程序将输入缓冲区复制到输出缓冲区，而不验证输入缓冲区的大小是否小于输出缓冲区的大小，从而导致缓冲区溢出。

### Description:

The program copies an input buffer to an output buffer without verifying that the size of the input buffer is less than the size of the output buffer, leading to a buffer overflow.

### 详细描述:

当程序试图将更多数据放入缓冲区而不是它可以容纳时，或者当程序试图将数据放入缓冲区边界之外的内存区域时，就会出现缓冲区溢出情况。最简单的错误类型和缓冲区溢出的最常见原因是“经典”情况，程序复制缓冲区而不限制复制的数量。存在其他变体，但经典溢出的存在强烈暗示程序员甚至不考虑最基本的安全保护。

### Extended Description:

A buffer overflow condition exists when a program attempts to put more data in a buffer than it can hold, or when a program attempts to put data in a memory area outside of the boundaries of a buffer. The simplest type of error, and the most common cause of buffer overflows, is the "classic" case in which the program copies the buffer without restricting how much is copied. Other variants exist, but the existence of a classic overflow strongly suggests that the programmer is not considering even the most basic of security protections.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在代码级别，基于堆栈和基于堆的溢出没有显着差异，因此通常不需要区分它们。从攻击者的角度来看，它们可能完全不同，因为开发它们需要不同的技术。

Weakness ID: 121

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Stack-based Buffer Overflow zh: -->基于堆栈的缓冲区溢出*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

基于堆栈的缓冲区溢出条件是被覆盖的缓冲区被分配在堆栈上的条件（即，是局部变量，或者很少是函数的参数）。

### Description:

A stack-based buffer overflow condition is a condition where the buffer being overwritten is allocated on the stack (i.e., is a local variable or, rarely, a parameter to a function).

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

执行堆栈上通常存在若干安全关键数据，这些数据可导致任意代码执行。最突出的是存储的返回地址，即当前函数执行完毕后执行应继续的存储器地址。攻击者可以使用攻击者也具有写访问权限的某些内存地址覆盖此值，并将任意代码放入其中以使用易受攻击程序的完全权限运行。或者，攻击者可以提供重要调用的地址，例如POSIX system（）调用，在堆栈上保留调用的参数。这通常被称为返回libc漏洞利用程序，因为攻击者通常会强制程序在返回时跳转到C标准库（libc）中的有趣例程。堆栈上常见的其他重要数据包括堆栈指针和帧指针，两个值指示计算内存地址的偏移量。修改这些值通常可以用于“写什么地方”条件。

### 笔记 (Notes):

基于堆栈的缓冲区溢出可以在返回地址覆盖，堆栈指针覆盖或帧指针覆盖中实例化。它们也可以被认为是函数指针覆盖，数组索引器覆盖或写入什么位置条件等。

Weakness ID: 122

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Heap-based Buffer Overflow zh: -->基于堆的缓冲区溢出*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

堆溢出条件是缓冲区溢出，其中可以覆盖的缓冲区在内存的堆部分中分配，通常意味着缓冲区是使用诸如malloc（）之类的例程分配的。

### Description:

A heap overflow condition is a buffer overflow, where the buffer that can be overwritten is allocated in the heap portion of memory, generally meaning that the buffer was allocated using a routine such as malloc().

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

基于堆的缓冲区溢出通常与基于堆栈的缓冲区溢出一样危险。

Weakness ID: 123

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Write-what-where Condition zh: -->写什么地方条件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

攻击者能够将任意值写入任意位置的任何情况，通常是缓冲区溢出的结果。

### Description:

Any condition where the attacker has the ability to write an arbitrary value to an arbitrary location, often as the result of a buffer overflow.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 124

提交日期 2009-10-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Buffer Underwrite ('Buffer Underflow') zh: -->缓冲区保护（'缓冲下溢'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件使用索引或指针写入缓冲区，该索引或指针在缓冲区开始之前引用内存位置。

### Description:

The software writes to a buffer using an index or pointer that references a memory location prior to the beginning of the buffer.

### 详细描述:

这通常发生在指针或其索引递减到缓冲区之前的位置，指针算术导致有效内存位置开始之前的位置或使用负索引时。

### Extended Description:

This typically occurs when a pointer or its index is decremented to a position before the buffer, when pointer arithmetic results in a position before the beginning of the valid memory location, or when a negative index is used.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能是由多个错误引起的，包括错误的偏移量或在缓冲区开始之前递减的数组索引（参见CWE-129）。

Weakness ID: 125

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Out-of-bounds Read zh: -->越界阅读*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件在预期缓冲区的结束或开始之前读取数据。

### Description:

The software reads data past the end, or before the beginning, of the intended buffer.

### 详细描述:

通常，这可能允许攻击者从其他内存位置读取敏感信息或导致崩溃。当代码读取可变数量的数据并假定存在用于停止读取操作的标记（例如字符串中的NUL）时，可能会发生崩溃。预期的哨兵可能不会位于越界内存中，导致过多的数据被读取，从而导致分段错误或缓冲区溢出。软件可以修改索引或执行引用超出缓冲区边界的存储器位置的指针算术。随后的读操作会产生未定义或意外的结果。

### Extended Description:

Typically, this can allow attackers to read sensitive information from other memory locations or cause a crash. A crash can occur when the code reads a variable amount of data and assumes that a sentinel exists to stop the read operation, such as a NUL in a string. The expected sentinel might not be located in the out-of-bounds memory, causinfg excessive data to be read, leading to a segmentation fault or a buffer overflow. The software may modify an index or perform pointer arithmetic that references a memory location that is outside of the boundaries of the buffer. A subsequent read operation then produces undefined or unexpected results.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

研究不足和报告不足。大多数问题可能被标记为缓冲区溢出。

Weakness ID: 126

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Buffer Over-read zh: -->缓冲区过度读取*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件使用缓冲区访问机制从缓冲区读取，例如在目标缓冲区之后引用内存位置的索引或指针。

### Description:

The software reads from a buffer using buffer access mechanisms such as indexes or pointers that reference memory locations after the targeted buffer.

### 详细描述:

这通常发生在指针或其索引递增到超出缓冲区边界的位置时，或者当指针算法导致有效内存位置之外的位置时，这一点通常会发生。这可能导致敏感信息的暴露或可能导致崩溃。

### Extended Description:

This typically occurs when the pointer or its index is incremented to a position beyond the bounds of the buffer or when pointer arithmetic results in a position outside of the valid memory location to name a few. This may result in exposure of sensitive information or possibly a crash.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这些问题可能是由于缺少标记值（CWE-463）或信任受用户影响的输入长度变量造成的。

Weakness ID: 127

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Buffer Under-read zh: -->缓冲区欠读*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件使用缓冲区访问机制从缓冲区读取，例如在目标缓冲区之前引用内存位置的索引或指针。

### Description:

The software reads from a buffer using buffer access mechanisms such as indexes or pointers that reference memory locations prior to the targeted buffer.

### 详细描述:

这通常发生在指针或其索引递减到缓冲区之前的位置，指针算术导致有效内存位置开始之前的位置，或者使用负索引时。这可能导致敏感信息的暴露或可能导致崩溃。

### Extended Description:

This typically occurs when the pointer or its index is decremented to a position before the buffer, when pointer arithmetic results in a position before the beginning of the valid memory location, or when a negative index is used. This may result in exposure of sensitive information or possibly a crash.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。

Weakness ID: 128

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Wrap-around Error zh: -->环绕错误*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

只要值增加超过其类型的最大值就会发生错误，因此会“绕回”到非常小的，负的或未定义的值。

### Description:

Wrap around errors occur whenever a value is incremented past the maximum value for its type and therefore "wraps around" to a very small, negative, or undefined value.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

由于计算机如何执行添加，如果基元增加超过其存储空间可能的最大值，系统将无法识别它，因此将每个位递增，就像它仍然有额外的空间一样。由于负数如何用二进制表示，解释为有符号的基元可以“包裹”到非常大的负值。

### 笔记 (Notes):

需要更仔细地检查溢出和环绕之间的关系，因为几个条目（包括CWE-190）密切相关。

Weakness ID: 129

提交日期 2009-10-29---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Validation of Array Index zh: -->数组索引的不正确验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在计算或使用数组索引时，产品使用不受信任的输入，但产品不验证或错误地验证索引以确保索引引用数组中的有效位置。

### Description:

The product uses untrusted input when calculating or using an array index, but the product does not validate or incorrectly validates the index to ensure the index references a valid position within the array.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

当使用大于数组大小的索引（例如JavaScript）时，这种弱点可以在不受控制的内存分配（CWE-789）中自动扩展数组。

Weakness ID: 130

提交日期 2008-09-09---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Length Parameter Inconsistency zh: -->长度参数不一致的处理不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件解析格式化的消息或结构，但它不处理或错误地处理与关联数据的实际长度不一致的长度字段。

### Description:

The software parses a formatted message or structure, but it does not handle or incorrectly handles a length field that is inconsistent with the actual length of the associated data.

### 详细描述:

如果攻击者可以操纵与输入相关联的长度参数，使其与输入的实际长度不一致，则可以利用这一点来使目标应用程序以意外的，可能是恶意的方式运行。这样做的一个可能动机是将任意大量的输入传递给应用程序。另一个可能的动机是通过为应用程序的后续属性包含无效数据来修改应用程序状态。这些弱点通常会导致诸如缓冲区溢出和任意代码执行之类的攻击。

### Extended Description:

If an attacker can manipulate the length parameter associated with an input such that it is inconsistent with the actual length of the input, this can be leveraged to cause the target application to behave in unexpected, and possibly, malicious ways. One of the possible motives for doing so is to pass in arbitrarily large input to the application. Another possible motivation is the modification of application state by including invalid data for subsequent properties of the application. Such weaknesses commonly lead to attacks such as buffer overflows and execution of arbitrary code.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能与其他类别重叠，包括零长度问题。

Weakness ID: 131

提交日期 2008-01-30---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Calculation of Buffer Size zh: -->缓冲区大小的计算不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件无法正确计算分配缓冲区时要使用的大小，这可能导致缓冲区溢出。

### Description:

The software does not correctly calculate the size to be used when allocating a buffer, which could lead to a buffer overflow.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这是一个广泛的类别。一些示例包括：简单的数学错误，错误地更新并行计数器，在将一个输入“转换”为另一种格式时不考虑大小差异（例如，URL规范化或其他可以生成比原始输入大的结果的转换，即“扩展” ）。公共报告中很少提供这种详细程度，因此很难找到好的例子。

Weakness ID: 132

提交日期 2008-09-09---> 修改日期 2009-10-29

* **Weakness Name:** *en: --> DEPRECATED (Duplicate): Miscalculated Null Termination zh: -->弃用（重复）：计算的空终止*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此条目已被弃用，因为它与CWE-170重复。所有内容均已转移至CWE-170。

### Description:

This entry has been deprecated because it was a duplicate of CWE-170. All content has been transferred to CWE-170.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 134

提交日期 2015-12-07---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Externally-Controlled Format String zh: -->使用外部控制的格式字符串*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用一个接受格式字符串作为参数的函数，但格式字符串来自外部源。

### Description:

The software uses a function that accepts a format string as an argument, but the format string originates from an external source.

### 详细描述:

当攻击者可以修改外部控制的格式字符串时，这可能导致缓冲区溢出，拒绝服务或数据表示问题。  
应该注意的是，在某些情况下，例如国际化，格式字符串集是由设计外部控制的。如果这些格式字符串的来源是可信的（例如，仅包含在只能由系统管理员修改的库文件中），则外部控件本身可能不会构成漏洞。

### Extended Description:

When an attacker can modify an externally-controlled format string, this can lead to buffer overflows, denial of service, or data representation problems.  
It should be noted that in some circumstances, such as internationalization, the set of format strings is externally controlled by design. If the source of these format strings is trusted (e.g. only contained in library files that are only modifiable by the system administrator), then the external control might not itself pose a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

任何支持格式字符串的编程语言都可能存在这种弱点。

Weakness ID: 135

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Calculation of Multi-Byte String Length zh: -->多字节字符串长度的计算不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法正确计算可包含宽字节或多字节字符的字符串长度。

### Description:

The software does not correctly calculate the length of strings that can contain wide or multi-byte characters.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 138

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Special Elements zh: -->特殊元素的中和不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和特殊元素，这些元素在被发送到下游组件时可被解释为控制元素或语法标记。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as control elements or syntactic markers when they are sent to a downstream component.

### 详细描述:

大多数语言和协议都有自己的特殊元素，如字符和保留字。这些特殊元素可以带来控制意义。如果软件不阻止外部控制或影响包含这些特殊元素，则程序的控制流程可能会改变。例如，Unix和Windows都将符号<（“小于”）解释为“从文件读取输入”。

### Extended Description:

Most languages and protocols have their own special elements such as characters and reserved words. These special elements can carry control implications. If software does not prevent external control or influence over the inclusion of such special elements, the control flow of the program may be altered from what was intended. For example, both Unix and Windows interpret the symbol < ("less than") as meaning "read input from a file".

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

当中间人的端点模型不考虑特定于协议的特殊元素时，这种弱点可能与中介（例如代理或应用防火墙）中的解释冲突或交互错误有关。

Weakness ID: 140

提交日期 2008-04-11---> 修改日期 2017-05-03

* **Weakness Name:** *en: --> Improper Neutralization of Delimiters zh: -->分隔符的中和不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会中和或错误地中和分隔符。

### Description:

The software does not neutralize or incorrectly neutralizes delimiters.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 141

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Parameter/Argument Delimiters zh: -->参数/参数分隔符的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件接收来自上游组件的输入，但它不会中和或不正确地中和可能在发送到下游组件时被解释为参数或参数分隔符的特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as parameter or argument delimiters when they are sent to a downstream component.

### 详细描述:

在解析数据时，注入/缺失/格式错误的分隔符可能会导致进程执行意外操作。

### Extended Description:

As data is parsed, an injected/absent/malformed delimiter may cause the process to take unexpected actions.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 142

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Value Delimiters zh: -->价值分隔符的中和不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和在将它们发送到下游组件时可被解释为值分隔符的特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as value delimiters when they are sent to a downstream component.

### 详细描述:

在解析数据时，注入/缺失/格式错误的分隔符可能会导致进程执行意外操作。

### Extended Description:

As data is parsed, an injected/absent/malformed delimiter may cause the process to take unexpected actions.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 143

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Record Delimiters zh: -->记录分隔符的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和在将它们发送到下游组件时可被解释为记录分隔符的特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as record delimiters when they are sent to a downstream component.

### 详细描述:

在解析数据时，注入/缺失/格式错误的分隔符可能会导致进程执行意外操作。

### Extended Description:

As data is parsed, an injected/absent/malformed delimiter may cause the process to take unexpected actions.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 144

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Neutralization of Line Delimiters zh: -->线分隔符的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和在将它们发送到下游组件时可被解释为行分隔符的特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as line delimiters when they are sent to a downstream component.

### 详细描述:

在解析数据时，注入/缺失/格式错误的分隔符可能会导致进程执行意外操作。

### Extended Description:

As data is parsed, an injected/absent/malformed delimiter may cause the process to take unexpected actions.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

根据使用的语言和语法，这可能与记录分隔符（CWE-143）相同。

Weakness ID: 145

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Section Delimiters zh: -->截面分隔符的不正确中和*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和在将它们发送到下游组件时可被解释为段分隔符的特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as section delimiters when they are sent to a downstream component.

### 详细描述:

在解析数据时，注入/缺失/格式错误的分隔符可能会导致进程执行意外操作。  
区段分隔符的一个示例是多部分MIME消息中的边界字符串。在许多情况下，双线分隔符可以用作节分隔符。

### Extended Description:

As data is parsed, an injected/absent/malformed delimiter may cause the process to take unexpected actions.  
One example of a section delimiter is the boundary string in a multipart MIME message. In many cases, doubled line delimiters can serve as a section delimiter.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

根据使用的语言和语法，这可能与记录分隔符（CWE-143）相同。

Weakness ID: 146

提交日期 2008-04-11---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Improper Neutralization of Expression/Command Delimiters zh: -->表达式/命令分隔符的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和在将它们发送到下游组件时可被解释为表达式或命令分隔符的特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as expression or command delimiters when they are sent to a downstream component.

### 详细描述:

在解析数据时，注入/缺失/格式错误的分隔符可能会导致进程执行意外操作。

### Extended Description:

As data is parsed, an injected/absent/malformed delimiter may cause the process to take unexpected actions.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

shell元字符（在CWE-150中涵盖）是可能需要中和的潜在定界符的一个示例。

Weakness ID: 147

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Input Terminators zh: -->输入终结器的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和在将它们发送到下游组件时可被解释为输入终止符的特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as input terminators when they are sent to a downstream component.

### 详细描述:

例如，“。”在SMTP中表示邮件消息数据的结束，而空字符可用于字符串的结尾。

### Extended Description:

For example, a "." in SMTP signifies the end of mail message data, whereas a null character can be used for the end of a string.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 148

提交日期 2008-01-30---> 修改日期 2017-05-03

* **Weakness Name:** *en: --> Improper Neutralization of Input Leaders zh: -->输入领导者的中和不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当主要字符或序列（“领导者”）丢失或格式错误时，或者当只允许一个领导者使用多个领导者时，应用程序无法正确处理。

### Description:

The application does not properly handle when a leading character or sequence ("leader") is missing or malformed, or if multiple leaders are used when only one should be allowed.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 149

提交日期 2008-01-30---> 修改日期 2017-05-03

* **Weakness Name:** *en: --> Improper Neutralization of Quoting Syntax zh: -->引用语法的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

注入应用程序的引号可用于危害系统。在解析数据时，引用的注入/不存在/重复/格式错误的使用可能导致进程采取意外操作。

### Description:

Quotes injected into an application can be used to compromise a system. As data are parsed, an injected/absent/duplicate/malformed use of quotes may cause the process to take unexpected actions.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 150

提交日期 2008-01-30---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Neutralization of Escape, Meta, or Control Sequences zh: -->转义，元或控制序列的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和可能被解释为转义，元或控制字符序列的特殊元素，当它们被发送到下游组件时。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as escape, meta, or control character sequences when they are sent to a downstream component.

### 详细描述:

在解析数据时，注入/缺失/格式错误的分隔符可能会导致进程执行意外操作。

### Extended Description:

As data is parsed, an injected/absent/malformed delimiter may cause the process to take unexpected actions.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 151

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Comment Delimiters zh: -->评论分隔符的不正当中和*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件接收来自上游组件的输入，但它不会中和或错误地中和特殊元素，这些元素在被发送到下游组件时可以被解释为注释分隔符。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as comment delimiters when they are sent to a downstream component.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 152

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Macro Symbols zh: -->宏符号的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件接收来自上游组件的输入，但它不会中和或不正确地中和可能在被发送到下游组件时被解释为宏符号的特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as macro symbols when they are sent to a downstream component.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。

Weakness ID: 153

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Substitution Characters zh: -->替换字符的中和不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和在将它们发送到下游组件时可被解释为替换字符的特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as substitution characters when they are sent to a downstream component.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。

Weakness ID: 154

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Variable Name Delimiters zh: -->变量名称分隔符的不正确中和*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和在将它们发送到下游组件时可被解释为变量名称分隔符的特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as variable name delimiters when they are sent to a downstream component.

### 详细描述:

在解析数据时，注入的分隔符可能会导致进程采取导致攻击的意外操作。示例：“$”表示环境变量。

### Extended Description:

As data is parsed, an injected delimiter may cause the process to take unexpected actions that result in an attack. Example: "$" for an environment variable.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。

Weakness ID: 155

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Wildcards or Matching Symbols zh: -->对通配符或匹配符号的不正确中和*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件接收来自上游组件的输入，但它不会中和或不正确地中和可能被解释为通配符或匹配符号的特殊元素，当它们被发送到下游组件时。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as wildcards or matching symbols when they are sent to a downstream component.

### 详细描述:

在解析数据时，注入的元素可能会导致进程执行意外操作。

### Extended Description:

As data is parsed, an injected element may cause the process to take unexpected actions.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。

Weakness ID: 156

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Whitespace zh: -->空白中和不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和在将它们发送到下游组件时可被解释为空白的特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could be interpreted as whitespace when they are sent to a downstream component.

### 详细描述:

这可以包括空格，标签等。

### Extended Description:

This can include space, tab, etc.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可以与其他分隔符或分隔符重叠。

Weakness ID: 157

提交日期 2008-04-11---> 修改日期 2017-05-03

* **Weakness Name:** *en: --> Failure to Sanitize Paired Delimiters zh: -->未能消除配对分隔符*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法正确处理用于标记一组实体的开头和结尾的字符，例如括号，括号和大括号。

### Description:

The software does not properly handle the characters that are used to mark the beginning and ending of a group of entities, such as parentheses, brackets, and braces.

### 详细描述:

配对分隔符可能包括：  
  
  
<和>尖括号  
（和）括号  
{和}大括号  
[和]方括号  
“ “ 双引号  
''单引号

### Extended Description:

Paired delimiters might include:  
  
  
< and > angle brackets  
( and ) parentheses  
{ and } braces  
[ and ] square brackets  
" " double quotes  
' ' single quotes

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。

Weakness ID: 158

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Null Byte or NUL Character zh: -->空字节或NUL字符的中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件接收来自上游组件的输入，但是当它们被发送到下游组件时，它不会中和或不正确地中和NUL字符或空字节。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes NUL characters or null bytes when they are sent to a downstream component.

### 详细描述:

在解析数据时，注入的NUL字符或空字节可能导致软件认为输入比实际更早地终止，或者导致输入被误解。然后，这可以用于注入在空字节之后发生的潜在危险输入，或者绕过验证例程和其他保护机制。

### Extended Description:

As data is parsed, an injected NUL character or null byte may cause the software to believe the input is terminated earlier than it actually is, or otherwise cause the input to be misinterpreted. This could then be used to inject potentially dangerous input that occurs after the null byte or otherwise bypass validation routines and other protection mechanisms.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能是多个解释错误，其他交互错误，文件名等效等因素。

Weakness ID: 159

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Failure to Sanitize Special Element zh: -->未能消毒特殊元素*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这种以攻击为重的类别中的弱点无法正确过滤和解释用户控制输入中的特殊元素，这些元素可能会对软件行为和完整性产生负面影响。

### Description:

Weaknesses in this attack-focused category do not properly filter and interpret special elements in user-controlled input which could cause adverse effect on the software behavior and integrity.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目的子项列表远未完成。

Weakness ID: 160

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Leading Special Elements zh: -->领导特殊元素的不正当中和*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和可能在发送到下游组件时以意外方式解释的前导特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes leading special elements that could be interpreted in unexpected ways when they are sent to a downstream component.

### 详细描述:

在解析数据时，不正确处理前导特殊元素可能会导致进程采取导致攻击的意外操作。

### Extended Description:

As data is parsed, improperly handled leading special elements may cause the process to take unexpected actions that result in an attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 161

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Multiple Leading Special Elements zh: -->多种主要特殊元素的中和不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和多个可能在发送到下游组件时以意外方式解释的前导特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes multiple leading special elements that could be interpreted in unexpected ways when they are sent to a downstream component.

### 详细描述:

在解析数据时，不正确地处理多个前导特殊元素可能会导致进程采取导致攻击的意外操作。

### Extended Description:

As data is parsed, improperly handled multiple leading special elements may cause the process to take unexpected actions that result in an attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 162

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Neutralization of Trailing Special Elements zh: -->尾随特殊元素的中和不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和可能在发送到下游组件时以意外方式解释的尾随特殊元素。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes trailing special elements that could be interpreted in unexpected ways when they are sent to a downstream component.

### 详细描述:

在解析数据时，不正确处理尾随特殊元素可能会导致进程采取导致攻击的意外操作。

### Extended Description:

As data is parsed, improperly handled trailing special elements may cause the process to take unexpected actions that result in an attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 163

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Multiple Trailing Special Elements zh: -->多尾随特殊元素的中和不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和多个尾随特殊元素，这些特殊元素在被发送到下游组件时可能以意外方式解释。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes multiple trailing special elements that could be interpreted in unexpected ways when they are sent to a downstream component.

### 详细描述:

在解析数据时，不正确地处理多个尾随特殊元素可能会导致进程采取导致攻击的意外操作。

### Extended Description:

As data is parsed, improperly handled multiple trailing special elements may cause the process to take unexpected actions that result in an attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 164

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Internal Special Elements zh: -->内部特殊元素的中和不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和内部特殊元素，这些特殊元素在被发送到下游组件时可能以意外方式解释。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes internal special elements that could be interpreted in unexpected ways when they are sent to a downstream component.

### 详细描述:

在解析数据时，不正确地处理内部特殊元素可能会导致进程采取导致攻击的意外操作。

### Extended Description:

As data is parsed, improperly handled internal special elements may cause the process to take unexpected actions that result in an attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 165

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of Multiple Internal Special Elements zh: -->多个内部特殊元素的中和不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但它不会中和或不正确地中和多个内部特殊元素，这些特殊元素在被发送到下游组件时可能以意外方式解释。

### Description:

The software receives input from an upstream component, but it does not neutralize or incorrectly neutralizes multiple internal special elements that could be interpreted in unexpected ways when they are sent to a downstream component.

### 详细描述:

在解析数据时，不正确地处理多个内部特殊元素可能会导致进程采取导致攻击的意外操作。

### Extended Description:

As data is parsed, improperly handled multiple internal special elements may cause the process to take unexpected actions that result in an attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 166

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Missing Special Element zh: -->丢失特殊元素的处理不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但是当缺少预期的特殊元素时，它不会处理或错误处理。

### Description:

The software receives input from an upstream component, but it does not handle or incorrectly handles when an expected special element is missing.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 167

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Additional Special Element zh: -->附加特殊元素的处理不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，但是当缺少其他意外特殊元素时，它不会处理或错误处理。

### Description:

The software receives input from an upstream component, but it does not handle or incorrectly handles when an additional unexpected special element is missing.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 168

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Inconsistent Special Elements zh: -->对不一致的特殊元素的处理不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当两个或多个特殊字符或保留字之间存在不一致时，软件不会处理。

### Description:

The software does not handle when an inconsistency exists between two or more special characters or reserved words.

### 详细描述:

此问题的一个示例是，如果配对字符以错误的顺序出现，或者特殊字符未正确嵌套。

### Extended Description:

An example of this problem would be if paired characters appear in the wrong order, or if the special characters are not properly nested.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 170

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Null Termination zh: -->不适当的空终止*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会终止或错误地终止具有空字符或等效终结符的字符串或数组。

### Description:

The software does not terminate or incorrectly terminates a string or array with a null character or equivalent terminator.

### 详细描述:

空终止错误经常以两种不同的方式发生。逐个错误可能导致空值被写入超出边界，从而导致溢出。或者，程序可能会错误地使用strncpy（）函数调用，这会阻止添加空终止符。其他情况也是可能的。

### Extended Description:

Null termination errors frequently occur in two different ways. An off-by-one error could cause a null to be written out of bounds, leading to an overflow. Or, a program could use a strncpy() function call incorrectly, which prevents a null terminator from being added at all. Other scenarios are possible.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

因素：这通常是由其他弱点引起的，例如逐个错误，但它可能是边界条件违规的主要因素，例如缓冲区溢出。在缓冲区溢出中，它可以充当假定不可变数据的扩展器。

Weakness ID: 172

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Encoding Error zh: -->编码错误*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法正确编码或解码数据，从而导致意外值。

### Description:

The software does not properly encode or decode the data, resulting in unexpected values.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

部分重叠路径遍历和等价弱点。

Weakness ID: 173

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Handling of Alternate Encoding zh: -->替代编码的处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当输入使用对发送输入的控制范围有效的备用编码时，软件无法正确处理。

### Description:

The software does not properly handle when an input uses an alternate encoding that is valid for the control sphere to which the input is being sent.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 174

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Double Decoding of the Same Data zh: -->双重解码相同的数据*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件对相同的输入进行两次解码，这可能限制在解码操作之间发生的任何保护机制的有效性。

### Description:

The software decodes the same input twice, which can limit the effectiveness of any protection mechanism that occurs in between the decoding operations.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可能未充分研究。

Weakness ID: 175

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Mixed Encoding zh: -->混合编码的处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当相同的输入使用多种不同（混合）编码时，软件无法正确处理。

### Description:

The software does not properly handle when the same input uses several different (mixed) encodings.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 176

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Unicode Encoding zh: -->Unicode编码处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当输入包含Unicode编码时，软件无法正确处理。

### Description:

The software does not properly handle when an input contains Unicode encoding.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 177

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Handling of URL Encoding (Hex Encoding) zh: -->URL编码处理不当（十六进制编码）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当全部或部分输入已经过URL编码时，软件无法正确处理。

### Description:

The software does not properly handle when all or part of an input has been URL encoded.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 178

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Case Sensitivity zh: -->案例敏感性处理不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在访问或确定资源属性时，软件无法正确解释区分大小写的差异，从而导致结果不一致。

### Description:

The software does not properly account for differences in case sensitivity when accessing or determining the properties of a resource, leading to inconsistent results.

### 详细描述:

处理不当的区分大小写的数据可能会导致一些可能的后果，包括：  
  
  
不区分大小写的密码减少了密钥空间的大小，使得暴力攻击更容易  
使用备用名称绕过过滤器或访问控制  
使用备用名称的多个解释错误。

### Extended Description:

Improperly handled case sensitive data can lead to several possible consequences, including:  
  
  
case-insensitive passwords reducing the size of the key space, making brute force attacks easier  
bypassing filters or access controls using alternate names  
multiple interpretation errors using alternate names.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这些可能在Windows和Mac环境中研究不足，其中文件名不区分大小写，因此需要进行涉及大小写的等效操作。

Weakness ID: 179

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incorrect Behavior Order: Early Validation zh: -->行为顺序不正确：早期验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件在应用修改输入的保护机制之前验证输入，这可能允许攻击者通过仅在修改后出现的危险输入绕过验证。

### Description:

The software validates input before applying protection mechanisms that modify the input, which could allow an attacker to bypass the validation via dangerous inputs that only arise after the modification.

### 详细描述:

在数据经过规范化和清理后，软件需要在适当的时间验证数据。早期验证易受各种操作的影响，这些操作会导致标准化和清洁产生的危险输入。

### Extended Description:

Software needs to validate data at the proper time, after data has been canonicalized and cleansed. Early validation is susceptible to various manipulations that result in dangerous inputs that are produced by canonicalization and cleansing.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这些错误主要在路径遍历漏洞中报告，但只要验证发生，该概念就适用。

Weakness ID: 180

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Behavior Order: Validate Before Canonicalize zh: -->行为顺序不正确：在Canonicalize之前验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件在规范化之前验证输入，这阻止软件检测在规范化步骤之后变为无效的数据。

### Description:

The software validates input before it is canonicalized, which prevents the software from detecting data that becomes invalid after the canonicalization step.

### 详细描述:

攻击者可以使用此方法绕过验证并发起攻击，从而暴露可能会被阻止的弱点，例如注入。

### Extended Description:

This can be used by an attacker to bypass the validation and launch attacks that expose weaknesses that would otherwise be prevented, such as injection.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这与其他类别重叠。

Weakness ID: 181

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Behavior Order: Validate Before Filter zh: -->行为顺序不正确：在过滤前验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件在过滤之前验证数据，这可以防止软件检测到过滤步骤后变为无效的数据。

### Description:

The software validates data before it has been filtered, which prevents the software from detecting data that becomes invalid after the filtering step.

### 详细描述:

攻击者可以使用此方法绕过验证并发起攻击，从而暴露可能会被阻止的弱点，例如注入。

### Extended Description:

This can be used by an attacker to bypass the validation and launch attacks that expose weaknesses that would otherwise be prevented, such as injection.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此类别可能未被充分研究。

Weakness ID: 182

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Collapse of Data into Unsafe Value zh: -->数据崩溃成不安全的价值*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件以某种方式过滤数据，导致数据被缩减或“折叠”为违反预期安全属性的不安全值。

### Description:

The software filters data in a way that causes it to be reduced or "collapsed" into an unsafe value that violates an expected security property.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠正则表达式，尽管实现可能不一定使用正则表达式。

Weakness ID: 183

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Permissive Whitelist zh: -->允许的白名单*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序使用可接受值的“白名单”，但白名单包含至少一个不安全值，从而导致产生的弱点。

### Description:

An application uses a "whitelist" of acceptable values, but the whitelist includes at least one unsafe value, leading to resultant weaknesses.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 184

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incomplete Blacklist zh: -->不完整的黑名单*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序使用禁止值的“黑名单”，但黑名单不完整。

### Description:

An application uses a "blacklist" of prohibited values, but the blacklist is incomplete.

### 详细描述:

如果将不完整的黑名单用作安全机制，则软件可能允许非预期的值传递到应用程序逻辑中。

### Extended Description:

If an incomplete blacklist is used as a security mechanism, then the software may allow unintended values to pass into the application logic.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

不完整的黑名单经常会产生缺陷。一些不完整的黑名单问题可能来自多个解释错误，例如：危险shell元字符的黑名单可能不包含只在一个特定shell中有意义的元字符，而不是所有元字符;或者XSS操作的黑名单可能会忽略一个Web浏览器支持的异常构造，而不是其他构建器。

Weakness ID: 185

提交日期 2008-09-09---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Incorrect Regular Expression zh: -->正则表达式不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件以导致数据不正确匹配或比较的方式指定正则表达式。

### Description:

The software specifies a regular expression in a way that causes data to be improperly matched or compared.

### 详细描述:

当正则表达式用于过滤或验证等保护机制时，这可能允许攻击者绕过对传入数据的预期限制。

### Extended Description:

When the regular expression is used in protection mechanisms such as filtering or validation, this may allow an attacker to bypass the intended restrictions on the incoming data.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

虽然与白名单/黑名单问题存在一些重叠，但此条目旨在处理错误编写的正则表达式，无论其预期用途如何。并非每个正则表达式都用作白名单或黑名单。此外，除了正则表达式之外，还可以使用其他机制来实现白名单和黑名单。

Weakness ID: 186

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Overly Restrictive Regular Expression zh: -->过度限制性的正则表达式*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

正则表达式过于严格，可以防止检测到危险值。

### Description:

A regular expression is overly restrictive, which prevents dangerous values from being detected.

### 详细描述:

这个弱点与正则表达式的复杂性无关。而是关于正则表达式与所有预期的值不匹配。考虑使用正则表达式将可接受的值列入白名单或将不需要的术语列入黑名单。过度限制性的正则表达式错过了一些潜在的安全相关值，导致误报\*或\*假阴性，这取决于在代码中如何使用正则表达式。考虑表达式/ [0-8] /其中意图是/ [0-9] /。这个表达式并不“复杂”，但是当程序员计划检查它时，值“9”不匹配。

### Extended Description:

This weakness is not about regexp complexity. Rather It is about a regular expression that does not match all values that are intended. Consider the use of a regexp to whitelist acceptable values or to blacklist unwanted terms. An overly restrictive regexp misses some potentially security-relevant values leading to either false positives \*or\* false negatives, depending on how the regexp is being used within the code. Consider the expression /[0-8]/ where the intention was /[0-9]/. This expression is not “complex” but the value “9” is not matched when maybe the programmer planned to check for it.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可能会重叠白名单/黑名单错误。

Weakness ID: 187

提交日期 2018-03-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Partial String Comparison zh: -->部分字符串比较*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件执行比较，该比较仅在确定是否存在匹配（例如子串）之前检查因子的一部分，从而导致产生的弱点。

### Description:

The software performs a comparison that only examines a portion of a factor before determining whether there is a match, such as a substring, leading to resultant weaknesses.

### 详细描述:

例如，攻击者可能通过提供与较大的正确密码的关联部分匹配的小密码来成功进行身份验证。

### Extended Description:

For example, an attacker might succeed in authentication by providing a small password that matches the associated portion of the larger, correct password.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这在概念上类似于其他弱点，例如验证不充分和正则表达式错误。这是一些弱点的主要原因。

Weakness ID: 188

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Reliance on Data/Memory Layout zh: -->依赖于数据/内存布局*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件对协议数据或内存如何在较低级别组织进行无效假设，从而导致意外的程序行为。

### Description:

The software makes invalid assumptions about how protocol data or memory is organized at a lower level, resulting in unintended program behavior.

### 详细描述:

更改平台或协议版本时，内存中的数据组织可能会以非预期的方式发生变化。例如，某些体系结构可能将局部变量A和B放在彼此旁边，A顶部;有些人可能会把它们放在一起，B顶在上面;和其他人可能会添加一些填充。填充大小可以变化以确保每个变量与适当的字大小对齐。  
在协议实现中，通常计算相对于另一个字段的偏移量以挑选出特定的数据。特殊情况（通常涉及新协议版本）可能会添加以不寻常方式更改数据布局的极端情况。结果可能是实现访问数据包中的非预期字段，将一种类型的数据视为另一种类型的数据。

### Extended Description:

When changing platforms or protocol versions, in-memory organization of data may change in unintended ways. For example, some architectures may place local variables A and B right next to each other with A on top; some may place them next to each other with B on top; and others may add some padding to each. The padding size may vary to ensure that each variable is aligned to a proper word size.  
In protocol implementations, it is common to calculate an offset relative to another field to pick out a specific piece of data. Exceptional conditions, often involving new protocol versions, may add corner cases that change the data layout in an unusual way. The result can be that an implementation accesses an unintended field in the packet, treating data of one type as data of another type.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 190

提交日期 2009-01-12---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Integer Overflow or Wraparound zh: -->整数溢出或环绕*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当逻辑假定结果值始终大于原始值时，软件执行可产生整数溢出或环绕的计算。当计算用于资源管理或执行控制时，这可能引入其他弱点。

### Description:

The software performs a calculation that can produce an integer overflow or wraparound, when the logic assumes that the resulting value will always be larger than the original value. This can introduce other weaknesses when the calculation is used for resource management or execution control.

### 详细描述:

当整数值递增到太大而无法存储在关联表示中的值时，会发生整数溢出或回绕。发生这种情况时，该值可能会换行变为非常小或负数。虽然这可能是在依赖包装的情况下的预期行为，但如果包装是意外的，则会产生安全性后果。如果可以使用用户提供的输入触发整数溢出，则尤其如此。当结果用于控制循环，做出安全决策或确定行为（例如内存分配，复制，连接等）中的偏移量或大小时，这将成为安全关键。

### Extended Description:

An integer overflow or wraparound occurs when an integer value is incremented to a value that is too large to store in the associated representation. When this occurs, the value may wrap to become a very small or negative number. While this may be intended behavior in circumstances that rely on wrapping, it can have security consequences if the wrap is unexpected. This is especially the case if the integer overflow can be triggered using user-supplied inputs. This becomes security-critical when the result is used to control looping, make a security decision, or determine the offset or size in behaviors such as memory allocation, copying, concatenation, etc.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

整数溢出可能是缓冲区溢出的主要溢出。

Weakness ID: 191

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Integer Underflow (Wrap or Wraparound) zh: -->整数下溢（换行或环绕）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品从另一个值中减去一个值，使得结果小于允许的最小整数值，这会产生一个不等于正确结果的值。

### Description:

The product subtracts one value from another, such that the result is less than the minimum allowable integer value, which produces a value that is not equal to the correct result.

### 详细描述:

这可能发生在签名和未签名的情况下。

### Extended Description:

This can happen in signed and unsigned cases.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。

Weakness ID: 192

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Integer Coercion Error zh: -->整数强制误差*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

整数强制是指与原始数据类型的类型转换，扩展或截断有关的一组缺陷。

### Description:

Integer coercion refers to a set of flaws pertaining to the type casting, extension, or truncation of primitive data types.

### 详细描述:

几个缺陷属于整数强制误差的范畴。在大多数情况下，这些错误本身只会导致可用性和数据完整性问题。但是，在某些情况下，它们可能会导致其他更复杂的安全相关缺陷，例如缓冲区溢出情况。

### Extended Description:

Several flaws fall under the category of integer coercion errors. For the most part, these errors in and of themselves result only in availability and data integrity issues. However, in some circumstances, they may result in other, more complicated security related flaws, such as buffer overflow conditions.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在C中，可能是“强制”在语义上与“强制转换”不同，可能取决于程序员是否直接指定转换，或者编译器是否隐式指定转换。这会影响此节点和其他节点（如CWE-681）的显示，以及是否存在足够的差异以便拆分这些节点。

Weakness ID: 193

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Off-by-one Error zh: -->一个错误*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品计算或使用的错误最大值或最小值比正确值多1或少1。

### Description:

A product calculates or uses an incorrect maximum or minimum value that is 1 more, or 1 less, than the correct value.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这并不总是缓冲区溢出。例如，逐个错误可能是部分比较中的因素，从错误的内存位置读取，不正确的条件等。

Weakness ID: 194

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Unexpected Sign Extension zh: -->意外的标志扩展*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件对一个数字执行操作，使其在转换为更大的数据类型时进行符号扩展。当原始数字为负数时，这会产生意外的值，从而导致产生的弱点。

### Description:

The software performs an operation on a number that causes it to be sign extended when it is transformed into a larger data type. When the original number is negative, this can produce unexpected values that lead to resultant weaknesses.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

签名扩展错误可能导致缓冲区溢出和其他基于内存的问题。它们也可能是其他弱点的因素，不是基于内存操作，而是依赖于数值计算。

Weakness ID: 195

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Signed to Unsigned Conversion Error zh: -->签名到无符号转换错误*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用带符号的基元并对无符号基元执行转换，如果无法使用无符号基元表示有符号基元的值，则可以产生意外值。

### Description:

The software uses a signed primitive and performs a cast to an unsigned primitive, which can produce an unexpected value if the value of the signed primitive can not be represented using an unsigned primitive.

### 详细描述:

依赖有符号和无符号数之间的隐式强制转换是危险的，因为结果可能会产生意外的值并违反程序所做的假设。  
通常，函数将返回负值以指示失败。当函数的结果用作大小参数时，使用这些负返回值可能会产生意外结果。例如，如果将负大小值传递给标准内存副本或分配函数，则它们将隐式转换为大的无符号值。这可能导致可利用的缓冲区溢出或下溢情况。

### Extended Description:

It is dangerous to rely on implicit casts between signed and unsigned numbers because the result can take on an unexpected value and violate assumptions made by the program.  
Often, functions will return negative values to indicate a failure. When the result of a function is to be used as a size parameter, using these negative return values can have unexpected results. For example, if negative size values are passed to the standard memory copy or allocation functions they will be implicitly cast to a large unsigned value. This may lead to an exploitable buffer overflow or underflow condition.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 196

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Unsigned to Signed Conversion Error zh: -->无符号转换为有符号转换错误*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用无符号基元并对有符号基元执行强制转换，如果无符号基元的值无法使用有符号基元表示，则可能会产生意外值。

### Description:

The software uses an unsigned primitive and performs a cast to a signed primitive, which can produce an unexpected value if the value of the unsigned primitive can not be represented using a signed primitive.

### 详细描述:

尽管问题比签名到无符号转换的频率低，但无符号到签名的转换可能是危险缓冲区保护条件的完美前提，这些条件允许攻击者向下移动堆栈，否则它们可能无法在正常的缓冲区溢出条件下访问。当大的无符号值转换为有符号值，然后用作缓冲区的索引或指针算术时，缓冲区承保经常发生。

### Extended Description:

Although less frequent an issue than signed-to-unsigned conversion, unsigned-to-signed conversion can be the perfect precursor to dangerous buffer underwrite conditions that allow attackers to move down the stack where they otherwise might not have access in a normal buffer overflow condition. Buffer underwrites occur frequently when large unsigned values are cast to signed values, and then used as indexes into a buffer or for pointer arithmetic.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 197

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Numeric Truncation Error zh: -->数字截断错误*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当基元转换为较小尺寸的基元并且数据在转换中丢失时，会发生截断错误。

### Description:

Truncation errors occur when a primitive is cast to a primitive of a smaller size and data is lost in the conversion.

### 详细描述:

当基元被转换为较小的基元时，大值的高位比特在转换中丢失，可能导致意外的值不等于原始值。可能需要此值作为缓冲区，循环迭代器或简单必需状态数据的索引。在任何情况下，该值都不可信，系统将处于未定义状态。尽管可以有效地使用该方法来隔离值的低位，但这种用法很少，并且截断通常意味着发生了实现错误。

### Extended Description:

When a primitive is cast to a smaller primitive, the high order bits of the large value are lost in the conversion, potentially resulting in an unexpected value that is not equal to the original value. This value may be required as an index into a buffer, a loop iterator, or simply necessary state data. In any case, the value cannot be trusted and the system will be in an undefined state. While this method may be employed viably to isolate the low bits of a value, this usage is rare, and truncation usually implies that an implementation error has occurred.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

尽管流行软件中的漏洞已在2008年和2009年发布，但传统上这种弱点一直未得到充分研究和报道不足。

Weakness ID: 198

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Incorrect Byte Ordering zh: -->使用不正确的字节顺序*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件接收来自上游组件的输入，但在处理输入时不考虑字节排序（例如，大端和小端），导致使用不正确的数字或​​值。

### Description:

The software receives input from an upstream component, but it does not account for byte ordering (e.g. big-endian and little-endian) when processing the input, causing an incorrect number or value to be used.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在报告的。

Weakness ID: 200

提交日期 2009-12-28---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Information Exposure zh: -->信息曝光*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

信息暴露是有意或无意地向未明确授权访问该信息的行为者披露信息。

### Description:

An information exposure is the intentional or unintentional disclosure of information to an actor that is not explicitly authorized to have access to that information.

### 详细描述:

信息要么：  
  
  
在产品自身的功能中被视为敏感的，例如私人消息;要么  
提供有关产品或其环境的信息，这些信息可能在攻击中有用，但攻击者通常无法使用，例如可远程访问的产品的安装路径。  
  
  
许多信息暴露是结果（例如，PHP脚本错误揭示了程序的完整路径），但它们也可能是主要的（例如加密中的时间差异）。有许多不同类型的问题涉及信息曝光。它们的严重程度可以根据所揭示的信息类型而广泛。

### Extended Description:

The information either:  
  
  
is regarded as sensitive within the product's own functionality, such as a private message; or  
provides information about the product or its environment that could be useful in an attack but is normally not available to the attacker, such as the installation path of a product that is remotely accessible.  
  
  
Many information exposures are resultant (e.g. PHP script error revealing the full path of the program), but they can also be primary (e.g. timing discrepancies in cryptography). There are many different types of problems that involve information exposures. Their severity can range widely depending on the type of information that is revealed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 201

提交日期 2010-09-27---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Sent Data zh: -->通过已发送数据的信息曝光*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

通过发送数据意外暴露敏感信息是指数据的传输，这些数据本身是敏感的，或者在通过标准数据信道进一步利用系统时是有用的。

### Description:

The accidental exposure of sensitive information through sent data refers to the transmission of data which are either sensitive in and of itself or useful in the further exploitation of the system through standard data channels.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 202

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Exposure of Sensitive Data Through Data Queries zh: -->通过数据查询暴露敏感数据*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在尝试保密信息时，攻击者通常可以使用统计信息来推断某些信息。

### Description:

When trying to keep information confidential, an attacker can often infer some of the information by using statistics.

### 详细描述:

在数据不应该绑定到单个用户的情况下，但是大量用户应该能够进行“擦除”用户身份的查询，则可以获得关于用户的信息 - 例如，通过指定搜索已知对该用户而言唯一的术语。

### Extended Description:

In situations where data should not be tied to individual users, but a large number of users should be able to make queries that "scrub" the identity of users, it may be possible to get information about a user -- e.g., by specifying search terms that are known to be unique to that user.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 203

提交日期 2009-12-28---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Discrepancy zh: -->信息暴露通过差异*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品的行为不同或以暴露与产品状态相关的安全相关信息的方式发送不同的响应，例如特定操作是否成功。

### Description:

The product behaves differently or sends different responses in a way that exposes security-relevant information about the state of the product, such as whether a particular operation was successful or not.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 204

提交日期 2010-09-27---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Response Discrepancy Information Exposure zh: -->响应差异信息暴露*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件以允许参与者确定该参与者的控制范围之外的系统状态信息的方式对传入请求提供不同的响应。

### Description:

The software provides different responses to incoming requests in a way that allows an actor to determine system state information that is outside of that actor's control sphere.

### 详细描述:

此问题在身份验证期间经常发生，其中失败登录消息的差异可能允许攻击者确定用户名是否有效。这些暴露可能是无意的（错误的）或有意的（设计）。

### Extended Description:

This issue frequently occurs during authentication, where a difference in failed-login messages could allow an attacker to determine if the username is valid or not. These exposures can be inadvertent (bug) or intentional (design).

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可以重叠与升级权限相关的错误

Weakness ID: 205

提交日期 2009-12-28---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Behavioral Discrepancy zh: -->通过行为差异进行信息曝光*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品的行为表明基于（1）产品的内部状态或（2）与同一类别中的其他产品的差异的重要差异。

### Description:

The product's actions indicate important differences based on (1) the internal state of the product or (2) differences from other products in the same class.

### 详细描述:

例如，OS指纹识别等攻击在很大程度上依赖于行为和响应差异。

### Extended Description:

For example, attacks such as OS fingerprinting rely heavily on both behavioral and response discrepancies.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 206

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure of Internal State Through Behavioral Inconsistency zh: -->通过行为不一致信息暴露内部状态*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品中的两个单独操作会导致产品以可被攻击者观察到的方式表现不同，并显示有关产品内部状态的安全相关信息，例如特定操作是否成功。

### Description:

Two separate operations in a product cause the product to behave differently in a way that is observable to an attacker and reveals security-relevant information about the internal state of the product, such as whether a particular operation was successful or not.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 207

提交日期 2009-12-28---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through an External Behavioral Inconsistency zh: -->信息暴露通过外部行为不一致*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品的行为与其他类似产品的行为不同，其方式是攻击者可以观察到并暴露与使用哪种产品相关的安全相关信息。

### Description:

The product behaves differently than other products like it, in a way that is observable to an attacker and exposes security-relevant information about which product is being used.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 208

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Timing Discrepancy zh: -->信息暴露通过时间差异*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品中的两个单独的操作需要不同的时间来完成，其方式是对于演员可观察到并且揭示关于产品状态的安全相关信息，例如特定操作是否成功。

### Description:

Two separate operations in a product require different amounts of time to complete, in a way that is observable to an actor and reveals security-relevant information about the state of the product, such as whether a particular operation was successful or not.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

通常在加密应用程序和算法中是主要的。

Weakness ID: 209

提交日期 2009-01-12---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Information Exposure Through an Error Message zh: -->信息通过错误消息曝光*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件会生成一条错误消息，其中包含有关其环境，用户或关联数据的敏感信息。

### Description:

The software generates an error message that includes sensitive information about its environment, users, or associated data.

### 详细描述:

敏感信息本身可能是有价值的信息（例如密码），或者对于发起其他更致命的攻击可能是有用的。如果攻击失败，攻击者可能会使用服务器提供的错误信息来启动另一个更集中的攻击。例如，尝试利用路径遍历弱点（CWE-22）可能会产生已安装应用程序的完整路径名。反过来，这可以用于选择适当数量的“..”序列以导航到目标文件。使用SQL注入（CWE-89）的攻击最初可能不会成功，但错误消息可能会显示格式错误的查询，这会暴露查询逻辑甚至是查询中使用的密码或其他敏感信息。

### Extended Description:

The sensitive information may be valuable information on its own (such as a password), or it may be useful for launching other, more deadly attacks. If an attack fails, an attacker may use error information provided by the server to launch another more focused attack. For example, an attempt to exploit a path traversal weakness (CWE-22) might yield the full pathname of the installed application. In turn, this could be used to select the proper number of ".." sequences to navigate to the targeted file. An attack using SQL injection (CWE-89) might not initially succeed, but an error message could reveal the malformed query, which would expose query logic and possibly even passwords or other sensitive information used within the query.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 210

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Self-generated Error Message zh: -->通过自生错误消息进行信息曝光*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件识别错误情况并创建自己的包含敏感信息的诊断或错误消息。

### Description:

The software identifies an error condition and creates its own diagnostic or error messages that contain sensitive information.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 211

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Externally-Generated Error Message zh: -->通过外部生成的错误消息进行信息暴露*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件执行触发外部诊断或错误消息的操作，该消息不是由软件直接生成的，例如由软件使用的编程语言解释器生成的错误。该错误可能包含敏感的系统信息。

### Description:

The software performs an operation that triggers an external diagnostic or error message that is not directly generated by the software, such as an error generated by the programming language interpreter that the software uses. The error can contain sensitive system information.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这本质上是由于产品内部的弱点或交互错误导致的漏洞。

Weakness ID: 212

提交日期 2009-12-28---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Cross-boundary Removal of Sensitive Data zh: -->不正确的跨境删除敏感数据*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用包含敏感数据的资源，但在与其他控制领域中的actor存储，传输或共享资源之前，它不会正确删除该数据。

### Description:

The software uses a resource that contains sensitive data, but it does not properly remove that data before it stores, transfers, or shares the resource with actors in another control sphere.

### 详细描述:

可能包含敏感数据的资源包括文档，数据包，消息，数据库等。虽然此数据可能对共享资源的单个用户或小组用户有用，但可能需要在资源可以在外部共享之前将其删除受信任的组织。去除过程有时称为清洁或擦洗。  
例如，用于编辑文档的软件可能不会删除敏感数据，例如审阅者注释或存储文档的本地路径名。或者，在向Internet站点发出传出请求之前，代理可能不会从标头中删除内部IP地址。

### Extended Description:

Resources that may contain sensitive data include documents, packets, messages, databases, etc. While this data may be useful to an individual user or small set of users who share the resource, it may need to be removed before the resource can be shared outside of the trusted group. The process of removal is sometimes called cleansing or scrubbing.  
For example, software that is used for editing documents might not remove sensitive data such as reviewer comments or the local pathname where the document is stored. Or, a proxy might not remove an internal IP address from headers before making an outgoing request to an Internet site.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目旨在与结果信息泄漏不同，包括因不正确的缓冲区初始化和重用，不正确的加密，交互错误和多个解释错误而导致的泄漏。此条目可能被视为隐私泄露，具体取决于泄露的信息类型。

Weakness ID: 213

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Intentional Information Exposure zh: -->故意信息曝光*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品的设计或配置明确要求发布可被管理员视为敏感的信息。

### Description:

A product's design or configuration explicitly requires the publication of information that could be regarded as sensitive by an administrator.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这与其他类别重叠，因为某些功能可能是开发人员想要的，但被用户或系统管理员视为弱点。在大多数情况下，它与CWE-209不同：信息通过错误消息曝光，因为CWE-209通常是无意的。

Weakness ID: 214

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Process Environment zh: -->通过流程环境进行信息曝光*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

使用敏感参数，环境变量或操作系统上其他进程可以看到的其他元素调用进程。

### Description:

A process is invoked with sensitive arguments, environment variables, or other elements that can be seen by other processes on the operating system.

### 详细描述:

许多操作系统允许用户列出有关其他用户拥有的进程的信息。此信息可能包括命令行参数或环境变量设置。当此数据包含敏感信息（如凭据）时，可能允许其他用户对软件或相关资源发起攻击。

### Extended Description:

Many operating systems allow a user to list information about processes that are owned by other users. This information could include command line arguments or environment variable settings. When this data contains sensitive information such as credentials, it might allow other users to launch an attack against the software or related resources.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

研究不足，特别是环境变量。

Weakness ID: 215

提交日期 2010-09-27---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Debug Information zh: -->通过调试信息暴露信息*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该应用程序包含调试代码，可以将敏感信息暴露给不受信任的各方。

### Description:

The application contains debugging code that can expose sensitive information to untrusted parties.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这与其他类别重叠。

Weakness ID: 216

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Containment Errors (Container Errors) zh: -->遏制错误（容器错误）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这试图涵盖在“容器”中包含不正确数据的各种问题。

### Description:

This tries to cover various problems in which improper data are included within a "container."

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目与其他与封装和权限相关的条目密切相关，最终可能证明是重复的。

Weakness ID: 217

提交日期 2009-05-27---> 修改日期 2009-10-29

* **Weakness Name:** *en: --> DEPRECATED: Failure to Protect Stored Data from Modification zh: -->已弃用：未能保护存储数据不被修改*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这种弱点已被弃用，因为它融合了许多弱点并使其混淆。以前在这个弱点中涵盖的问题可以在CWE-766和CWE-767中找到。

### Description:

This weakness has been deprecated because it incorporated and confused multiple weaknesses. The issues formerly covered in this weakness can be found at CWE-766 and CWE-767.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 218

提交日期 2008-09-09---> 修改日期 2009-10-29

* **Weakness Name:** *en: --> DEPRECATED (Duplicate): Failure to provide confidentiality for stored data zh: -->已弃用（重复）：未对存储的数据提供机密性*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这个弱点已被弃用，因为它与CWE-493重复。所有内容均已转移至CWE-493。

### Description:

This weakness has been deprecated because it was a duplicate of CWE-493. All content has been transferred to CWE-493.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 219

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Sensitive Data Under Web Root zh: -->Web Root下的敏感数据*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序将敏感数据存储在Web文档根目录下，访问控制不足，这可能使不受信任的各方可以访问它们。

### Description:

The application stores sensitive data under the web document root with insufficient access control, which might make it accessible to untrusted parties.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 220

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Sensitive Data Under FTP Root zh: -->FTP根目录下的敏感数据*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序将敏感数据存储在FTP文档根目录下，访问控制不足，这可能使不受信任方可以访问它。

### Description:

The application stores sensitive data under the FTP document root with insufficient access control, which might make it accessible to untrusted parties.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

由于使用chroot，各种Unix FTP服务器需要FTP根目录下的密码文件。

### 笔记 (Notes):

没有笔记

Weakness ID: 221

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Loss or Omission zh: -->信息丢失或遗漏*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会记录或不正确地记录导致错误决策或妨碍以后分析的安全相关信息。

### Description:

The software does not record, or improperly records, security-relevant information that leads to an incorrect decision or hampers later analysis.

### 详细描述:

这可能是结果，例如，缓冲区溢出可能会在产品记录事件之前触发崩溃。

### Extended Description:

This can be resultant, e.g. a buffer overflow might trigger a crash before the product can log the event.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 222

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Truncation of Security-relevant Information zh: -->截断安全相关信息*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序以可能模糊攻击来源或性质的方式截断安全相关信息的显示，记录或处理。

### Description:

The application truncates the display, recording, or processing of security-relevant information in a way that can obscure the source or nature of an attack.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 223

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Omission of Security-relevant Information zh: -->省略与安全相关的信息*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序不会记录或显示对于识别攻击的来源或性质或确定某个操作是否安全非常重要的信息。

### Description:

The application does not record or display information that would be important for identifying the source or nature of an attack, or determining if an action is safe.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 224

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Obscured Security-relevant Information by Alternate Name zh: -->通过替代名称隐藏安全相关信息*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件根据受影响实体的备用名称而不是规范名称记录与安全相关的信息。

### Description:

The software records security-relevant information according to an alternate name of the affected entity, instead of the canonical name.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 225

提交日期 I`m don`t know---> 修改日期 2009-10-29

* **Weakness Name:** *en: --> DEPRECATED (Duplicate): General Information Management Problems zh: -->弃用（重复）：一般信息管理问题*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这个弱点可以在CWE-199找到。

### Description:

This weakness can be found at CWE-199.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 226

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Sensitive Information Uncleared Before Release zh: -->敏感信息在发布前未清除*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在将该资源提供给另一个控制领域的一方之前，该软件不能完全清除数据结构，文件或其他资源中以前使用的信息。

### Description:

The software does not fully clear previously used information in a data structure, file, or other resource, before making that resource available to a party in another control sphere.

### 详细描述:

这通常是由新数据产生的，这些新数据不像旧数据那样长，这使得旧数据的一部分仍然可用。在数据长度可变但相关数据结构不可变的其他情况下，可能会发生等效错误。如果在使用后未清除内存，则可能允许非预期的actor在重新分配内存时读取数据。

### Extended Description:

This typically results from new data that is not as long as the old data, which leaves portions of the old data still available. Equivalent errors can occur in other situations where the length of data is variable but the associated data structure is not. If memory is not cleared after use, it may allow unintended actors to read the data when the memory is reallocated.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

CWE-226与CWE-212之间存在密切关联。差异部分是透视。 CWE-226面向资源生命周期的最后阶段，资源被删除，删除，过期或以其他方式发布以供重用。从技术上讲，这涉及转移到不同的控制领域，其中资源的原始内容不再相关。但是，CWE-212适用于有意与他人共享的资源中的敏感数据，因此它们仍处于活动状态。从CWE研究视角（CWE-1000）的角度来看，这种区别很有用。

Weakness ID: 228

提交日期 2008-09-09---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Syntactically Invalid Structure zh: -->句法无效结构的处理不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品不处理或错误处理与相关规范相关的语法不完整的输入。

### Description:

The product does not handle or incorrectly handles input that is not syntactically well-formed with respect to the associated specification.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目需要更多调查。公共漏洞研究通常侧重于产生无效结构的操纵，而不是那些操纵所利用的弱点。例如，常见攻击涉及发出省略必填字段的请求，这可能会在某些情况下触发崩溃。崩溃可能是由于一个命名链，如CWE-690（未经检查的返回值为空指针解除引用），但公开报告很少涉及漏洞的这一方面。

Weakness ID: 229

提交日期 2008-04-11---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Improper Handling of Values zh: -->不正确的价值处理*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果输入中未提供参数，字段或参数的预期数量的值，或者这些值未定义，则软件无法正确处理。

### Description:

The software does not properly handle when the expected number of values for parameters, fields, or arguments is not provided in input, or if those values are undefined.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 230

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Handling of Missing Values zh: -->错误值的处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

指定参数，字段或参数名称时，软件不处理或错误处理，但缺少关联值，即它为空，空白或空。

### Description:

The software does not handle or incorrectly handles when a parameter, field, or argument name is specified, but the associated value is missing, i.e. it is empty, blank, or null.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

一些“端口扫描崩溃”错误可能是由于这一点，但缺乏诊断使得很难确定。

Weakness ID: 231

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Extra Values zh: -->对额外值的处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当提供的值多于预期时，软件不会处理或错误处理。

### Description:

The software does not handle or incorrectly handles when more values are provided than expected.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能会重叠缓冲区溢出。

Weakness ID: 232

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Handling of Undefined Values zh: -->未定义值的处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果未为相关参数，字段或参数名称定义或支持值，则软件不会处理或错误处理。

### Description:

The software does not handle or incorrectly handles when a value is not defined or supported for the associated parameter, field, or argument name.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 233

提交日期 2013-07-17---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Improper Handling of Parameters zh: -->参数处理不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果输入中未提供预期数量的参数，字段或参数，或者这些参数未定义，则软件无法正确处理。

### Description:

The software does not properly handle when the expected number of parameters, fields, or arguments is not provided in input, or if those parameters are undefined.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 234

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Failure to Handle Missing Parameter zh: -->无法处理缺失的参数*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果向函数发送的参数太少，该函数仍会从堆栈中弹出预期数量的参数。潜在地，可以在函数中耗尽可变数量的参数。

### Description:

If too few arguments are sent to a function, the function will still pop the expected number of arguments from the stack. Potentially, a variable number of arguments could be exhausted in a function as well.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目将在CWE的未来版本中弃用。术语“缺失参数”用于PLOVER和CLASP，具有完全不同的含义。但是，来自两个分类法的数据已合并到此条目中。在PLOVER中，它旨在覆盖不包含必需参数的格式错误的输入，例如CGI请求中缺少的参数。此条目的观察示例和分类来自PLOVER。然而，描述，说明性示例和其他信息源自CLASP。它们与函数参数的数量不正确相关，CWE-685已经涵盖了这些参数。

Weakness ID: 235

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Extra Parameters zh: -->额外参数的处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当具有相同名称的参数，字段或参数的数量超过预期量时，软件不会处理或错误处理。

### Description:

The software does not handle or incorrectly handles when the number of parameters, fields, or arguments with the same name exceeds the expected amount.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此类问题在多个解释漏洞和各种HTTP攻击中起着重要作用。

Weakness ID: 236

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Undefined Parameters zh: -->未定义参数的处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当产品未定义或支持特定参数，字段或参数名称时，软件不会处理或错误处理。

### Description:

The software does not handle or incorrectly handles when a particular parameter, field, or argument name is not defined or supported by the product.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 237

提交日期 2009-03-10---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Improper Handling of Structural Elements zh: -->结构元素处理不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不处理或错误处理与复杂结构相关的输入。

### Description:

The software does not handle or incorrectly handles inputs that are related to complex structures.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 238

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Incomplete Structural Elements zh: -->对不完整结构元素的处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当未完全指定特定结构元素时，软件不处理或错误处理。

### Description:

The software does not handle or incorrectly handles when a particular structural element is not completely specified.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可以是其他问题的主要原因。

Weakness ID: 239

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Failure to Handle Incomplete Element zh: -->未能处理不完整的元素*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果未完全指定特定元素，则软件无法正确处理。

### Description:

The software does not properly handle when a particular element is not completely specified.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 240

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Inconsistent Structural Elements zh: -->对不一致结构元素的处理不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当两个或多个结构元素应该一致时，软件不会处理或错误处理，但不是。

### Description:

The software does not handle or incorrectly handles when two or more structural elements should be consistent, but are not.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 241

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Handling of Unexpected Data Type zh: -->不正确处理意外数据类型*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当特定元素不是预期类型时，软件不处理或错误处理，例如它需要一个数字（0-9），但提供一个字母（A-Z）。

### Description:

The software does not handle or incorrectly handles when a particular element is not the expected type, e.g. it expects a digit (0-9) but is provided with a letter (A-Z).

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可能未充分研究。

Weakness ID: 242

提交日期 2008-01-30---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Inherently Dangerous Function zh: -->使用固有危险的功能*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序调用一个永远不能保证安全工作的函数。

### Description:

The program calls a function that can never be guaranteed to work safely.

### 详细描述:

无论如何使用，某些功能都会以危险的方式运行。通常在不考虑安全问题的情况下实施此类别中的功能。 gets（）函数是不安全的，因为它不对其输入的大小执行边界检查。攻击者可以轻松地将任意大小的输入发送到gets（）并溢出目标缓冲区。类似地，在读取静态分配的字符数组时，>> operator不安全，因为它不对输入的大小执行边界检查。攻击者可以轻松地将任意大小的输入发送到>>运算符并溢出目标缓冲区。

### Extended Description:

Certain functions behave in dangerous ways regardless of how they are used. Functions in this category were often implemented without taking security concerns into account. The gets() function is unsafe because it does not perform bounds checking on the size of its input. An attacker can easily send arbitrarily-sized input to gets() and overflow the destination buffer. Similarly, the >> operator is unsafe to use when reading into a statically-allocated character array because it does not perform bounds checking on the size of its input. An attacker can easily send arbitrarily-sized input to the >> operator and overflow the destination buffer.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 243

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Creation of chroot Jail Without Changing Working Directory zh: -->在不改变工作目录的情况下创建chroot监狱*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序使用chroot（）系统调用来创建一个jail，但之后不会更改工作目录。这不会阻止访问jail之外的文件。

### Description:

The program uses the chroot() system call to create a jail, but does not change the working directory afterward. This does not prevent access to files outside of the jail.

### 详细描述:

chroot（）的不当使用可能允许攻击者逃离chroot监狱。 chroot（）函数调用不会更改进程的当前工作目录，因此在调用chroot（）之后，相对路径仍可能引用chroot jail之外的文件系统资源。

### Extended Description:

Improper use of chroot() may allow attackers to escape from the chroot jail. The chroot() function call does not change the process's current working directory, so relative paths may still refer to file system resources outside of the chroot jail after chroot() has been called.

### 问题背景 (Background Detail):

chroot（）系统调用允许进程改变其对文件系统根目录的感知。正确调用chroot（）之后，进程无法访问新根目录定义的目录树之外的任何文件。这种环境称为chroot jail，通常用于防止进程被破坏并用于访问未授权文件的可能性。例如，许多FTP服务器在chroot jails中运行，以防止发现服务器中新漏洞的攻击者能够下载系统上的密码文件或其他敏感文件。

### 笔记 (Notes):

没有笔记

Weakness ID: 244

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Clearing of Heap Memory Before Release ('Heap Inspection') zh: -->释放前不正确清除堆内存（'堆检查'）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

使用realloc（）来调整存储敏感信息的缓冲区可能会使敏感信息暴露于攻击，因为它不会从内存中删除。

### Description:

Using realloc() to resize buffers that store sensitive information can leave the sensitive information exposed to attack, because it is not removed from memory.

### 详细描述:

如果未从内存中删除敏感数据（如密码或加密密钥），则可能会使用“堆检查”攻击向攻击者公开，该攻击使用内存转储或其他方法读取敏感数据。 realloc（）函数通常用于增加已分配内存块的大小。此操作通常需要将旧存储器块的内容复制到新的更大的块中。此操作使原始块的内容保持不变但程序无法访问，从而阻止程序从内存中擦除敏感数据。如果攻击者以后可以检查内存转储的内容，则可能会暴露敏感数据。

### Extended Description:

When sensitive data such as a password or an encryption key is not removed from memory, it could be exposed to an attacker using a "heap inspection" attack that reads the sensitive data using memory dumps or other methods. The realloc() function is commonly used to increase the size of a block of allocated memory. This operation often requires copying the contents of the old memory block into a new and larger block. This operation leaves the contents of the original block intact but inaccessible to the program, preventing the program from being able to scrub sensitive data from memory. If an attacker can later examine the contents of a memory dump, the sensitive data could be exposed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 245

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> J2EE Bad Practices: Direct Management of Connections zh: -->J2EE不良做法：直接管理连接*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

J2EE应用程序直接管理连接，而不是使用容器的连接管理工具。

### Description:

The J2EE application directly manages connections, instead of using the container's connection management facilities.

### 详细描述:

J2EE标准禁止直接管理连接。它要求应用程序使用容器的资源管理工具来获取与资源的连接。每个主要Web应用程序容器都提供池化数据库连接管理作为其资源管理框架的一部分。在应用程序中复制此功能很困难且容易出错，这是J2EE标准禁止的部分原因。

### Extended Description:

The J2EE standard forbids the direct management of connections. It requires that applications use the container's resource management facilities to obtain connections to resources. Every major web application container provides pooled database connection management as part of its resource management framework. Duplicating this functionality in an application is difficult and error prone, which is part of the reason it is forbidden under the J2EE standard.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 246

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> J2EE Bad Practices: Direct Use of Sockets zh: -->J2EE不良做法：直接使用套接字*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

J2EE应用程序直接使用套接字而不是使用框架方法调用。

### Description:

The J2EE application directly uses sockets instead of using framework method calls.

### 详细描述:

当没有更高级别的协议可用时，J2EE标准仅允许将套接字用于与遗留系统通信的目的。编写自己的通信协议需要解决棘手的安全问题。  
如果没有安全专家的严格审查，自定义通信协议可能会遇到安全问题。许多相同的问题适用于标准协议的自定义实现。虽然通常有更多资源可用于解决与实施标准协议相关的安全问题，但攻击者也可以使用这些资源。

### Extended Description:

The J2EE standard permits the use of sockets only for the purpose of communication with legacy systems when no higher-level protocol is available. Authoring your own communication protocol requires wrestling with difficult security issues.  
Without significant scrutiny by a security expert, chances are good that a custom communication protocol will suffer from security problems. Many of the same issues apply to a custom implementation of a standard protocol. While there are usually more resources available that address security concerns related to implementing a standard protocol, these resources are also available to attackers.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 247

提交日期 2008-04-11---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> DEPRECATED (Duplicate): Reliance on DNS Lookups in a Security Decision zh: -->弃用（重复）：依赖于安全决策中的DNS查找*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此条目已被弃用，因为它与CWE-350重复。所有内容均已转移至CWE-350。

### Description:

This entry has been deprecated because it was a duplicate of CWE-350. All content has been transferred to CWE-350.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 248

提交日期 2008-01-30---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Uncaught Exception zh: -->未捕获的异常*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

从函数抛出异常，但它没有被捕获。

### Description:

An exception is thrown from a function, but it is not caught.

### 详细描述:

未捕获异常时，可能会导致程序崩溃或泄露敏感信息。

### Extended Description:

When an exception is not caught, it may cause the program to crash or expose sensitive information.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 249

提交日期 2009-07-27---> 修改日期 2009-10-29

* **Weakness Name:** *en: --> DEPRECATED: Often Misused: Path Manipulation zh: -->弃用：经常被滥用：路径操纵*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

由于名称混淆以及多个弱点的偶然组合，此条目已被弃用。其大部分内容已转移至CWE-785。

### Description:

This entry has been deprecated because of name confusion and an accidental combination of multiple weaknesses. Most of its content has been transferred to CWE-785.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目已弃用，原因有几个。主要原因是“路径操纵”术语和描述的过载。此条目的原始描述与原始七恶毒王国论文中“经常滥用：文件系统”项目的描述相同。然而，Seven Pernicious Kingdoms还有一个“Path Manipulation”短语用于外部控制路径名（CWE-73），这是符号链接跟踪和路径遍历的一个因素，7PK中都没有明确提到这一点。 Fortify使用短语“经常滥用：路径操作”来解决更广泛的问题，通常用于与缓冲区管理相关的问题。鉴于此术语的多次冲突使用，CWE用户可能错误地映射到此条目。弃用的第二个原因是缓冲区处理函数中的多个弱点的隐含组合。此条目的重点通常是路径转换函数及其与缓冲区溢出的关联。但是，Fortify的一些Vulncat条目具有术语“路径操作”，但描述了一个非溢出的弱点，其中缓冲区不能保证包含整个路径名，即有信息截断（参见CWE-222的类似概念） 。可以在CWE的未来版本中创建此非溢出弱点的新条目。

Weakness ID: 250

提交日期 2008-01-30---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Execution with Unnecessary Privileges zh: -->执行不必要的权限*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件以高于所需最低级别的权限级别执行操作，这会产生新的弱点或放大其他弱点的后果。

### Description:

The software performs an operation at a privilege level that is higher than the minimum level required, which creates new weaknesses or amplifies the consequences of other weaknesses.

### 详细描述:

可能会暴露新的弱点，因为使用额外的特权（例如root或Administrator）运行可能会禁用操作系统或周围环境执行的正常安全检查。如果在提升权限下运行时，其他预先存在的弱点可能会变成安全漏洞。  
权限管理功能可以以一些不太明显的方式运行，并且它们在不同平台上具有不同的怪癖。如果您从一个非root用户转换到另一个非root用户，则这些不一致性尤其明显。信号处理程序和生成的进程在拥有进程的特权下运行，因此如果在信号触发或执行子进程时进程以root身份运行，则信号处理程序或子进程将以root权限运行。

### Extended Description:

New weaknesses can be exposed because running with extra privileges, such as root or Administrator, can disable the normal security checks being performed by the operating system or surrounding environment. Other pre-existing weaknesses can turn into security vulnerabilities if they occur while operating at raised privileges.  
Privilege management functions can behave in some less-than-obvious ways, and they have different quirks on different platforms. These inconsistencies are particularly pronounced if you are transitioning from one non-root user to another. Signal handlers and spawned processes run at the privilege of the owning process, so if a process is running as root when a signal fires or a sub-process is executed, the signal handler or sub-process will operate with root privileges.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

与CWE-653（权限分离不足）密切相关。 CWE-653是为每个权限提供单独的组件; CWE-250旨在确保每个组件具有尽可能少的权限。

Weakness ID: 252

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Unchecked Return Value zh: -->未选中的返回值*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件不检查方法或函数的返回值，这可以防止它检测到意外的状态和条件。

### Description:

The software does not check the return value from a method or function, which can prevent it from detecting unexpected states and conditions.

### 详细描述:

两个常见的程序员假设是“这个函数调用永远不会失败”和“这个函数调用失败并不重要”。如果攻击者可以强制该函数失败或以其他方式返回不期望的值，则后续程序逻辑可能导致漏洞，因为该软件不处于程序员假定的状态。例如，如果程序调用函数来删除权限但不检查返回代码以确保成功删除权限，则程序将继续以更高权限运行。

### Extended Description:

Two common programmer assumptions are "this function call can never fail" and "it doesn't matter if this function call fails". If an attacker can force the function to fail or otherwise return a value that is not expected, then the subsequent program logic could lead to a vulnerability, because the software is not in a state that the programmer assumes. For example, if the program calls a function to drop privileges but does not check the return code to ensure that privileges were successfully dropped, then the program will continue to operate with the higher privileges.

### 问题背景 (Background Detail):

许多功能将为其行动的成功返回一些价值。这将提醒程序是否处理由该功能引起的任何错误。

### 笔记 (Notes):

没有笔记

Weakness ID: 253

提交日期 2009-03-10---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Check of Function Return Value zh: -->函数返回值检查错误*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件错误地检查函数的返回值，这会阻止软件检测错误或异常情况。

### Description:

The software incorrectly checks a return value from a function, which prevents the software from detecting errors or exceptional conditions.

### 详细描述:

重要和常见的功能将为其行动的成功返回一些价值。这将提醒程序是否处理由该功能引起的任何错误。

### Extended Description:

Important and common functions will return some value about the success of its actions. This will alert the program whether or not to handle any errors caused by that function.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 256

提交日期 2008-01-30---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Unprotected Storage of Credentials zh: -->不受保护的凭证存储*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

以明文存储密码可能会导致系统受损。

### Description:

Storing a password in plaintext may result in a system compromise.

### 详细描述:

当密码以纯文本形式存储在应用程序的属性或配置文件中时，会出现密码管理问题。在配置文件中存储明文密码允许任何能够读取该文件的人访问受密码保护的资源。

### Extended Description:

Password management issues occur when a password is stored in plaintext in an application's properties or configuration file. Storing a plaintext password in a configuration file allows anyone who can read the file access to the password-protected resource.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 257

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Storing Passwords in a Recoverable Format zh: -->以可恢复的格式存储密码*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

以可恢复格式存储密码使得它们受到恶意用户的密码重用攻击。实际上，应该注意的是，可恢复的加密密码与明文密码相比没有明显的好处，因为它们不仅可以被恶意攻击者重用，还可以被恶意内部人员重用。如果系统管理员可以直接恢复密码，或对可用信息使用强力搜索，则管理员可以在其他帐户上使用密码。

### Description:

The storage of passwords in a recoverable format makes them subject to password reuse attacks by malicious users. In fact, it should be noted that recoverable encrypted passwords provide no significant benefit over plaintext passwords since they are subject not only to reuse by malicious attackers but also by malicious insiders. If a system administrator can recover a password directly, or use a brute force search on the available information, the administrator can use the password on other accounts.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

需要更密切地研究这个节点的含义，特别是关于“可恢复”的含义。

Weakness ID: 258

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Empty Password in Configuration File zh: -->配置文件中的空密码*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

使用空字符串作为密码是不安全的。

### Description:

Using an empty string as a password is insecure.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 259

提交日期 2010-02-16---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Hard-coded Password zh: -->使用硬编码密码*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个硬编码密码，用于自己的入站身份验证或外部组件的出站通信。

### Description:

The software contains a hard-coded password, which it uses for its own inbound authentication or for outbound communication to external components.

### 详细描述:

硬编码密码通常会导致严重的身份验证失败，系统管理员很难检测到。一旦检测到，就很难修复，因此管理员可能会被迫完全禁用该产品。主要有两种变化：  
  
入站：该软件包含一个检查硬编码密码的身份验证机制。  
出站：软件连接到另一个系统或组件，它包含用于连接到该组件的硬编码密码。  
  
在入站变体中，将创建默认管理帐户，并将简单密码硬编码到产品中并与该帐户关联。此硬编码密码对于产品的每次安装都是相同的，并且系统管理员通常无法在不手动修改程序或修补软件的情况下更改或禁用密码。如果密码被发现或发布（在Internet上很常见），那么任何知道此密码的人都可以访问该产品。最后，由于软件的所有安装都具有相同的密码，即使在不同的组织中，也可以实现诸如蠕虫之类的大规模攻击。  
Outbound变体适用于使用后端服务进行身份验证的前端系统。后端服务可能需要一个可以轻松发现的固定密码。程序员可以简单地将这些后端凭证硬编码到前端软件中。该程序的任何用户都可以提取密码。具有硬编码密码的客户端系统构成了更大的威胁，因为从二进制文件中提取密码通常非常简单。

### Extended Description:

A hard-coded password typically leads to a significant authentication failure that can be difficult for the system administrator to detect. Once detected, it can be difficult to fix, so the administrator may be forced into disabling the product entirely. There are two main variations:  
  
Inbound: the software contains an authentication mechanism that checks for a hard-coded password.  
Outbound: the software connects to another system or component, and it contains hard-coded password for connecting to that component.  
  
In the Inbound variant, a default administration account is created, and a simple password is hard-coded into the product and associated with that account. This hard-coded password is the same for each installation of the product, and it usually cannot be changed or disabled by system administrators without manually modifying the program, or otherwise patching the software. If the password is ever discovered or published (a common occurrence on the Internet), then anybody with knowledge of this password can access the product. Finally, since all installations of the software will have the same password, even across different organizations, this enables massive attacks such as worms to take place.  
The Outbound variant applies to front-end systems that authenticate with a back-end service. The back-end service may require a fixed password which can be easily discovered. The programmer may simply hard-code those back-end credentials into the front-end software. Any user of that program may be able to extract the password. Client-side systems with hard-coded passwords pose even more of a threat, since the extraction of a password from a binary is usually very simple.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目可能应分为多个变体：入站变体（如第二个演示示例中所示）和出站变体（如第一个演示示例中所示）。这些变体可能会产生不同的后果，可检测性等。请参阅扩展说明。

Weakness ID: 260

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Password in Configuration File zh: -->配置文件中的密码*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件将密码存储在配置文件中，该配置文件可能对不知道密码的演员可访问。

### Description:

The software stores a password in a configuration file that might be accessible to actors who do not know the password.

### 详细描述:

这可能导致使用密码的系统受到损害。攻击者可以访问此文件并了解存储的密码，或者更糟糕的是，将密码更改为他们选择的密码。

### Extended Description:

This can result in compromise of the system for which the password is used. An attacker could gain access to this file and learn the stored password or worse yet, change the password to one of their choosing.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 261

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Weak Cryptography for Passwords zh: -->密码弱密码学*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

使用简单编码隐藏密码不会保护密码。

### Description:

Obscuring a password with a trivial encoding does not protect the password.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

当密码以纯文本形式存储在应用程序的属性或配置文件中时，会出现密码管理问题。程序员可以通过使用编码功能（例如base 64编码）来模糊密码来尝试解决密码管理问题，但这种努力不能充分保护密码。 “crypt”系列函数使用弱加密算法，应该避免使用。它可能存在于某些项目中以实现兼容性。

Weakness ID: 262

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Not Using Password Aging zh: -->不使用密码时效*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果没有适当的机制来管理密码老化，用户将无法及时更新密码。

### Description:

If no mechanism is in place for managing password aging, users will have no incentive to update passwords in a timely manner.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 263

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Password Aging with Long Expiration zh: -->长期过期的密码时效*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

允许密码老化未经检查可能导致密码完整性降低。

### Description:

Allowing password aging to occur unchecked can result in the possibility of diminished password integrity.

### 详细描述:

正如忽略包含管理密码老化的功能是危险的，因此允许密码老化继续未选中。密码必须具有最长寿命，之后用户需要使用新的不同密码进行更新。

### Extended Description:

Just as neglecting to include functionality for the management of password aging is dangerous, so is allowing password aging to continue unchecked. Passwords must be given a maximum life span, after which a user is required to update with a new and different password.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 266

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Privilege Assignment zh: -->权限分配不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品错误地将特权分配给特定的actor，为该actor创建一个非预期的控制范围。

### Description:

A product incorrectly assigns a privilege to a particular actor, creating an unintended sphere of control for that actor.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 267

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Privilege Defined With Unsafe Actions zh: -->使用不安全操作定义的权限*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

特定的权限，角色，功能或权限可用于执行非预期的不安全操作，即使将其分配给正确的实体也是如此。

### Description:

A particular privilege, role, capability, or right can be used to perform unsafe actions that were not intended, even when it is assigned to the correct entity.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这与授权和访问控制问题重叠。

Weakness ID: 268

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Privilege Chaining zh: -->特权链接*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

可以以允许实体执行不允许没有该组合的不安全动作的方式组合两个不同的特权，角色，功能或权限。

### Description:

Two distinct privileges, roles, capabilities, or rights can be combined in a way that allows an entity to perform unsafe actions that would not be allowed without that combination.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

与Unsafe Privilege存在一些概念上的重叠。

Weakness ID: 269

提交日期 2008-09-09---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Privilege Management zh: -->权限管理不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件未正确分配，修改，跟踪或检查演员的权限，从而为该演员创建了一个非预期的控制范围。

### Description:

The software does not properly assign, modify, track, or check privileges for an actor, creating an unintended sphere of control for that actor.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

特权，权限和参与者（例如用户和组）之间的关系需要在Research视图中进一步细化。一个复杂因素是这些概念适用于两个不同的支柱，涉及资源控制（CWE-664）和保护机制故障（CWE-396）。

Weakness ID: 270

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Privilege Context Switching Error zh: -->特权上下文切换错误*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当软件在具有不同权限或控制范围的不同上下文之间切换时，该软件无法正确管理权限。

### Description:

The software does not properly manage privileges while it is switching between different contexts that have different privileges or spheres of control.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这个概念需要更多的研究。

Weakness ID: 271

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Privilege Dropping / Lowering Errors zh: -->权限下降/降低错误*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在将资源控制权交给没有这些权限的actor之前，该软件不会删除权限。

### Description:

The software does not drop privileges before passing control of a resource to an actor that does not have those privileges.

### 详细描述:

在某些情况下，使用提升权限执行的系统将切换进程/文件/等。到另一个进程或用户。如果未降低实体的权限，则提升的权限会在整个系统中传播，并可能传播给攻击者。

### Extended Description:

In some contexts, a system executing with elevated permissions will hand off a process/file/etc. to another process or user. If the privileges of an entity are not reduced, then elevated privileges are spread throughout a system and possibly to an attacker.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

CWE-271，CWE-272和CWE-250都密切相关，可能重叠。 CWE-271可能更适合作为一个类别。

Weakness ID: 272

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Least Privilege Violation zh: -->最低权限违规*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

执行操作后，应立即删除执行chroot（）等操作所需的提升权限级别。

### Description:

The elevated privilege level required to perform operations such as chroot() should be dropped immediately after the operation is performed.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

CWE-271，CWE-272和CWE-250都密切相关，可能重叠。 CWE-271可能更适合作为一个类别。

Weakness ID: 273

提交日期 2009-03-10---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Check for Dropped Privileges zh: -->不正确检查已删除的权限*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件尝试删除权限，但不检查或错误检查以查看删除是否成功。

### Description:

The software attempts to drop privileges but does not check or incorrectly checks to see if the drop succeeded.

### 详细描述:

如果删除失败，软件将继续使用提升的权限运行，这可能会为非特权用户提供额外的访问权限。

### Extended Description:

If the drop fails, the software will continue to run with the raised privileges, which might provide additional access to unprivileged users.

### 问题背景 (Background Detail):

在具有访问控制的基于Windows的环境中，使用模拟，以便可以由具有更高权限的服务器对客户端标识执行访问检查。通过模拟客户端，服务器被限制为客户端级别的安全性 - 尽管在不同的线程中它可能具有更高的权限。

### 笔记 (Notes):

没有笔记

Weakness ID: 274

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Insufficient Privileges zh: -->权利不足的处理不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当软件没有足够的权限执行操作时，软件不会处理或错误处理，从而导致产生的缺陷。

### Description:

The software does not handle or incorrectly handles when it has insufficient privileges to perform an operation, leading to resultant weaknesses.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠权限丢失，权限不足。

Weakness ID: 276

提交日期 2009-05-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Default Permissions zh: -->默认权限不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

安装后，该软件为将其公开给非预期的actor的对象设置了不正确的权限。

### Description:

The software, upon installation, sets incorrect permissions for an object that exposes it to an unintended actor.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 277

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Insecure Inherited Permissions zh: -->不安全的继承权限*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品定义一组不安全的权限，这些权限由程序创建的对象继承。

### Description:

A product defines a set of insecure permissions that are inherited by objects that are created by the program.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 278

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Insecure Preserved Inherited Permissions zh: -->不安全的保留继承权限*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品继承了一组对象的不安全权限，例如从存档文件复制时，没有用户意识或参与。

### Description:

A product inherits a set of insecure permissions for an object, e.g. when copying from an archive file, without user awareness or involvement.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 279

提交日期 2009-05-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Execution-Assigned Permissions zh: -->执行分配的权限不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在执行时，软件以违反用户指定的预期权限的方式设置对象的权限。

### Description:

While it is executing, the software sets the permissions of an object in a way that violates the intended permissions that have been specified by the user.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 280

提交日期 2009-03-10---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Insufficient Permissions or Privileges zh: -->对权限或权限不足的处理不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当应用程序没有足够的权限访问其权限所指定的资源或功能时，应用程序不会处理或错误处理。这可能导致它遵循可能使应用程序处于无效状态的意外代码路径。

### Description:

The application does not handle or incorrectly handles when it has insufficient privileges to access resources or functionality as specified by their permissions. This may cause it to follow unexpected code paths that may leave the application in an invalid state.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可以是主要的和结果。在主要时，它可能会暴露各种弱点，因为资源可能没有预期的状态，后续操作可能会失败。它通常来自未经检查的错误条件（CWE-391）。

Weakness ID: 281

提交日期 2009-05-27---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Preservation of Permissions zh: -->权限保护不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在复制，还原或共享对象时，该软件不保留权限或错误地保留权限，这可能导致它们具有比预期更少的限制权限。

### Description:

The software does not preserve permissions or incorrectly preserves permissions when copying, restoring, or sharing objects, which can cause them to have less restrictive permissions than intended.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 282

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Ownership Management zh: -->所有权管理不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件分配了错误的所有权，或者没有正确验证对象或资源的所有权。

### Description:

The software assigns the wrong ownership, or does not properly verify the ownership, of an object or resource.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

特权，权限和参与者（例如用户和组）之间的关系需要在Research视图中进一步细化。一个复杂因素是这些概念适用于两个不同的支柱，涉及资源控制（CWE-664）和保护机制故障（CWE-396）。

Weakness ID: 283

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Unverified Ownership zh: -->未经证实的所有权*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件未正确验证关键资源是否由适当的实体拥有。

### Description:

The software does not properly verify that a critical resource is owned by the proper entity.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这与比较，验证错误，权限和权限不足重叠。

Weakness ID: 284

提交日期 2008-09-09---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Access Control zh: -->访问控制不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不限制或不正确地限制对未授权演员的资源访问。

### Description:

The software does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

### 详细描述:

访问控制涉及使用多种保护机制，例如：  
  
身份验证（证明演员的身份）  
授权（确保给定的actor可以访问资源），以及  
问责制（跟踪已执行的活动）  
  
当任何机制未应用或以其他方式失败时，攻击者可以通过获取权限，读取敏感信息，执行命令，逃避检测等来危害软件的安全性。  
有两种不同的行为可能会引入访问控制缺陷：  
  
  
规范：为用户或资源明确指定了不正确的权限，权限，所有权等（例如，将密码文件设置为可全局写入，或​​为访客用户提供管理员功能）。此操作可由程序或管理员执行。  
强制执行：该机制包含的错误会阻止它正确执行指定的访问控制要求（例如，允许用户指定自己的权限，或允许语法错误的ACL产生不安全的设置）。此问题发生在程序本身内，因为它实际上并未强制执行管理员指定的预期安全策略。

### Extended Description:

Access control involves the use of several protection mechanisms such as:  
  
Authentication (proving the identity of an actor)  
Authorization (ensuring that a given actor can access a resource), and  
Accountability (tracking of activities that were performed)  
  
When any mechanism is not applied or otherwise fails, attackers can compromise the security of the software by gaining privileges, reading sensitive information, executing commands, evading detection, etc.  
There are two distinct behaviors that can introduce access control weaknesses:  
  
  
Specification: incorrect privileges, permissions, ownership, etc. are explicitly specified for either the user or the resource (for example, setting a password file to be world-writable, or giving administrator capabilities to a guest user). This action could be performed by the program or the administrator.  
Enforcement: the mechanism contains errors that prevent it from properly enforcing the specified access control requirements (e.g., allowing the user to specify their own privileges, or allowing a syntactically-incorrect ACL to produce insecure settings). This problem occurs within the program itself, in that it does not actually enforce the intended security policy that the administrator specifies.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这个项目需要更多的工作。可能的子类别包括：\*可信组包括不需要的实体（部分由CWE-286覆盖）\*组可以执行不需要的操作\* ACL解析错误不会失败关闭

Weakness ID: 285

提交日期 2009-01-12---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Authorization zh: -->授权不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当actor尝试访问资源或执行操作时，该软件不执行或错误地执行授权检查。

### Description:

The software does not perform or incorrectly performs an authorization check when an actor attempts to access a resource or perform an action.

### 详细描述:

假设具有给定身份的用户，授权是基于用户的权限和适用于资源的任何权限或其他访问控制规范来确定该用户是否可以访问给定资源的过程。  
当访问控制检查未一致地应用 - 或根本不应用 - 用户能够访问数据或执行不应被允许执行的操作。这可能导致各种各样的问题，包括信息泄露，拒绝服务和任意代码执行。

### Extended Description:

Assuming a user with a given identity, authorization is the process of determining whether that user can access a given resource, based on the user's privileges and any permissions or other access-control specifications that apply to the resource.  
When access control checks are not applied consistently - or not at all - users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information exposures, denial of service, and arbitrary code execution.

### 问题背景 (Background Detail):

访问控制列表（ACL）表示对给定对象具有权限的人员/内容。不同的操作系统以不同的方式实现（ACL）。在UNIX中，有三种类型的权限：读取，写入和执行。用户分为三个类用于文件访问：所有者，组所有者以及每个类具有单独权限集的所有其他用户。在Windows NT中，文件的权限有四种基本类型：“无访问权限”，“读取权限”，“更改访问权限”和“完全控制权”。 Windows NT扩展了UNIX中三种类型用户的概念，包括用户和组列表及其相关权限。用户可以创建对象（文件）并为该对象分配指定的权限。

### 笔记 (Notes):

没有笔记

Weakness ID: 286

提交日期 2008-09-09---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incorrect User Management zh: -->用户管理不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法在其环境中正确管理用户。

### Description:

The software does not properly manage a user within its environment.

### 详细描述:

可以将用户分配给错误的权限组（类），从而导致对敏感对象的非预期访问权限。

### Extended Description:

Users can be assigned to the wrong group (class) of permissions resulting in unintended access rights to sensitive objects.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

特权，权限和参与者（例如用户和组）之间的关系需要在Research视图中进一步细化。一个复杂因素是这些概念适用于与资源控制（CWE-664）和保护机制故障（CWE-693）相关的两个不同支柱。

Weakness ID: 287

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Authentication zh: -->身份验证不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当演员声称拥有特定身份时，该软件不会证明或证明该声明是正确的。

### Description:

When an actor claims to have a given identity, the software does not prove or insufficiently proves that the claim is correct.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能是由SQL注入漏洞和其他问题引起的。

Weakness ID: 288

提交日期 2008-09-09---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Authentication Bypass Using an Alternate Path or Channel zh: -->使用备用路径或通道进行身份验证旁路*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品需要身份验证，但产品具有不需要身份验证的备用路径或通道。

### Description:

A product requires authentication, but the product has an alternate path or channel that does not require authentication.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠未受保护的备用频道

Weakness ID: 289

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Authentication Bypass by Alternate Name zh: -->身份验证绕过备用名称*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件基于正在访问的资源的名称或执行访问的actor的名称执行身份验证，但它没有正确检查该资源或actor的所有可能名称。

### Description:

The software performs authentication based on the name of a resource being accessed, or the name of the actor performing the access, but it does not properly check all possible names for that resource or actor.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠等效编码，规范化，授权，多尾随斜杠，尾随空格，混合大小写和其他等效问题。

Weakness ID: 290

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Authentication Bypass by Spoofing zh: -->通过欺骗进行身份验证绕过*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这种以攻击为中心的弱点是由不正确实施的受欺骗攻击的身份验证方案引起的。

### Description:

This attack-focused weakness is caused by improperly implemented authentication schemes that are subject to spoofing attacks.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能是由于验证不充分造成的。

Weakness ID: 291

提交日期 2013-07-17---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Reliance on IP Address for Authentication zh: -->依赖于身份验证的IP地址*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用IP地址进行身份验证。

### Description:

The software uses an IP address for authentication.

### 详细描述:

IP地址很容易被欺骗。攻击者可以伪造他们发送的数据包的源IP地址，但响应数据包将返回伪造的IP地址。要查看响应数据包，攻击者必须嗅探受害者计算机和伪造IP地址之间的流量。为了完成所需的嗅探，攻击者通常会尝试将自己定位在与受害者计算机相同的子网上。攻击者可以通过使用源路由来规避此要求，但是今天在大部分Internet上都禁用了源路由。总之，IP地址验证可以是身份验证方案的有用部分，但它不应该是身份验证所需的单一因素。

### Extended Description:

IP addresses can be easily spoofed. Attackers can forge the source IP address of the packets they send, but response packets will return to the forged IP address. To see the response packets, the attacker has to sniff the traffic between the victim machine and the forged IP address. In order to accomplish the required sniffing, attackers typically attempt to locate themselves on the same subnet as the victim machine. Attackers may be able to circumvent this requirement by using source routing, but source routing is disabled across much of the Internet today. In summary, IP address verification can be a useful part of an authentication scheme, but it should not be the single factor required for authentication.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 292

提交日期 2013-07-17---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> DEPRECATED (Duplicate): Trusting Self-reported DNS Name zh: -->DEPRECATED（重复）：信任自我报告的DNS名称*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此条目已被弃用，因为它与CWE-350重复。所有内容均已转移至CWE-350。

### Description:

This entry has been deprecated because it was a duplicate of CWE-350. All content has been transferred to CWE-350.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 293

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Using Referer Field for Authentication zh: -->使用Referer字段进行身份验证*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

HTTP请求中的referer字段可以很容易地修改，因此不是消息完整性检查的有效方法。

### Description:

The referer field in HTTP requests can be easily modified and, as such, is not a valid means of message integrity checking.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

HTML请求中的referer字段可以被恶意用户简单地修改，使其无法用作检查相关请求的有效性的手段。

### 笔记 (Notes):

没有笔记

Weakness ID: 294

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Authentication Bypass by Capture-replay zh: -->身份验证绕过Capture-replay*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当软件设计使恶意用户可以通过将其重播到相关服务器以获得与原始消息相同的效果（或稍作更改）来嗅探网络流量并绕过身份验证时，存在捕获重放缺陷。

### Description:

A capture-replay flaw exists when the design of the software makes it possible for a malicious user to sniff network traffic and bypass authentication by replaying it to the server in question to the same effect as the original message (or with minor changes).

### 详细描述:

捕获重放攻击很常见，如果没有加密技术，很难打败。它们是网络注入攻击的一个子集，它依赖于观察先前发送的有效命令，然后在必要时稍微更改它们并将相同的命令重新发送到服务器。

### Extended Description:

Capture-replay attacks are common and can be difficult to defeat without cryptography. They are a subset of network injection attacks that rely on observing previously-sent valid commands, then changing them slightly if necessary and resending the same commands to the server.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 295

提交日期 2013-02-21---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Certificate Validation zh: -->证书验证不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不验证或错误验证证书。

### Description:

The software does not validate, or incorrectly validates, a certificate.

### 详细描述:

当证书无效或恶意时，它可能允许攻击者通过使用中间人（MITM）攻击欺骗可信实体。该软件可能连接到恶意主机，同时认为它是可信主机，或者软件可能被欺骗接受看似来自可信主机的欺骗数据。

### Extended Description:

When a certificate is invalid or malicious, it might allow an attacker to spoof a trusted entity by using a man-in-the-middle (MITM) attack. The software might connect to a malicious host while believing it is a trusted host, or the software might be deceived into accepting spoofed data that appears to originate from a trusted host.

### 问题背景 (Background Detail):

证书是将身份（主体）与加密密钥相关联的令牌。证书可用于检查公钥是否属于假定的所有者。

### 笔记 (Notes):

没有笔记

Weakness ID: 296

提交日期 2009-03-10---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Following of a Certificate's Chain of Trust zh: -->不正确地遵循证书的信任链*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不遵循或错误地遵循证书的信任链回到受信任的根证书，从而导致与该证书关联的任何资源的不正确信任。

### Description:

The software does not follow, or incorrectly follows, the chain of trust for a certificate back to a trusted root certificate, resulting in incorrect trust of any resource that is associated with that certificate.

### 详细描述:

如果系统不遵循证书对根服务器的信任链，则证书将失去作为信任度量的所有有用性。从本质上讲，从证书中获得的信任来自信任链 - 在该列表的末尾有一个信誉良好的可信实体。最终用户必须信任该信誉良好的来源，并且这个信誉良好的来源必须通过证书媒介担保相关资源。  
在某些情况下，这种信任遍历了几个互相担保的实体。最终用户信任的实体位于此信任链的一端，而证书挥舞资源位于链的另一端。如果用户在其中一个信任链的末尾收到证书，然后继续仅检查链中的第一个链接，则不会导出真正的信任，因为必须遍历整个链到可信来源以验证证书。  
可以通过多种方式破坏信任链，包括但不限于：  
  
  
链中的任何证书都是自签名的，除非它是根。  
并非每个中间证书都会被检查，从原始证书一直到根证书。  
中间的CA签名证书没有预期的基本约束或其他重要扩展。  
根证书已被泄露或授权给错误的一方。

### Extended Description:

If a system does not follow the chain of trust of a certificate to a root server, the certificate loses all usefulness as a metric of trust. Essentially, the trust gained from a certificate is derived from a chain of trust -- with a reputable trusted entity at the end of that list. The end user must trust that reputable source, and this reputable source must vouch for the resource in question through the medium of the certificate.  
In some cases, this trust traverses several entities who vouch for one another. The entity trusted by the end user is at one end of this trust chain, while the certificate-wielding resource is at the other end of the chain. If the user receives a certificate at the end of one of these trust chains and then proceeds to check only that the first link in the chain, no real trust has been derived, since the entire chain must be traversed back to a trusted source to verify the certificate.  
There are several ways in which the chain of trust might be broken, including but not limited to:  
  
  
Any certificate in the chain is self-signed, unless it the root.  
Not every intermediate certificate is checked, starting from the original certificate all the way up to the root certificate.  
An intermediate, CA-signed certificate does not have the expected Basic Constraints or other important extensions.  
The root certificate has been compromised or authorized to the wrong party.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 297

提交日期 2009-03-10---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Validation of Certificate with Host Mismatch zh: -->主机不匹配的证书验证不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件与提供证书的主机通信，但该软件未正确确保证书实际与该主机关联。

### Description:

The software communicates with a host that provides a certificate, but the software does not properly ensure that the certificate is actually associated with that host.

### 详细描述:

即使证书格式良好，签名并遵循信任链，它也可能只是与软件交互的站点不同的站点的有效证书。如果未正确检查证书的特定于主机的数据（例如主题中的公用名（CN）或X.509证书的主题备用名称（SAN）扩展名），则可能会出现重定向或欺骗攻击允许具有有效证书的恶意主机提供数据，模拟可信主机。为了确保数据完整性，证书必须有效，并且必须与正在访问的站点相关。  
即使软件试图检查主机名，仍然可能错误地检查主机名。例如，攻击者可以创建一个名称以可信名称开头，后跟NUL字节的证书，这可能导致某些基于字符串的比较仅检查包含受信任名称的部分。  
即使软件使用证书固定，如果软件在固定证书时未验证主机名，也会出现此弱点。

### Extended Description:

Even if a certificate is well-formed, signed, and follows the chain of trust, it may simply be a valid certificate for a different site than the site that the software is interacting with. If the certificate's host-specific data is not properly checked - such as the Common Name (CN) in the Subject or the Subject Alternative Name (SAN) extension of an X.509 certificate - it may be possible for a redirection or spoofing attack to allow a malicious host with a valid certificate to provide data, impersonating a trusted host. In order to ensure data integrity, the certificate must be valid and it must pertain to the site that is being accessed.  
Even if the software attempts to check the hostname, it is still possible to incorrectly check the hostname. For example, attackers could create a certificate with a name that begins with a trusted name followed by a NUL byte, which could cause some string-based comparisons to only examine the portion that contains the trusted name.  
This weakness can occur even when the software uses Certificate Pinning, if the software does not verify the hostname at the time a certificate is pinned.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 298

提交日期 2009-03-10---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Validation of Certificate Expiration zh: -->证书过期的验证不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

证书过期未经过验证或未正确验证，因此可能会将信任分配给因年龄而被放弃的证书。

### Description:

A certificate expiration is not validated or is incorrectly validated, so trust may be assigned to certificates that have been abandoned due to age.

### 详细描述:

如果不考虑证书的到期，则不一定通过它传达信任。因此，无法验证证书的有效性，并且证书的所有好处都将丢失。

### Extended Description:

When the expiration of a certificate is not taken into account, no trust has necessarily been conveyed through it. Therefore, the validity of the certificate cannot be verified and all benefit of the certificate is lost.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 299

提交日期 2009-03-10---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Check for Certificate Revocation zh: -->不正确的检查证书撤销*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不检查或错误地检查证书的撤销状态，这可能导致它使用已被泄露的证书。

### Description:

The software does not check or incorrectly checks the revocation status of a certificate, which may cause it to use a certificate that has been compromised.

### 详细描述:

对证书撤销进行不正确的检查是一个比相关证书失败更严重的缺陷。这是因为使用任何撤销的证书几乎肯定是恶意的。证书撤销的最常见原因是有问题的系统遭到破坏，结果是没有合法的服务器将使用撤销的证书，除非它们非常不同步。

### Extended Description:

An improper check for certificate revocation is a far more serious flaw than related certificate failures. This is because the use of any revoked certificate is almost certainly malicious. The most common reason for certificate revocation is compromise of the system in question, with the result that no legitimate servers will be using a revoked certificate, unless they are sorely out of sync.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 300

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Channel Accessible by Non-Endpoint ('Man-in-the-Middle') zh: -->非端点可访问的频道（'中间人'）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品不能充分验证通信信道两端的参与者的身份，或者不能充分确保信道的完整性，从而允许信道被非端点的参与者访问或影响。

### Description:

The product does not adequately verify the identity of actors at both ends of a communication channel, or does not adequately ensure the integrity of the channel, in a way that allows the channel to be accessed or influenced by an actor that is not an endpoint.

### 详细描述:

为了在双方之间建立安全通信，充分验证通信信道每端的实体身份通常很重要。验证不充分或不一致可能导致通信实体的识别不充分或不正确。这可能会产生负面影响，例如信道另一端的实体信任错位。攻击者可以通过插入通信实体和伪装成原始实体来利用它。在没有足够的身份验证的情况下，这样的攻击者可以窃听并可能修改原始实体之间的通信。

### Extended Description:

In order to establish secure communication between two parties, it is often important to adequately verify the identity of entities at each end of the communication channel. Inadequate or inconsistent verification may result in insufficient or incorrect identification of either communicating entity. This can have negative consequences such as misplaced trust in the entity at the other end of the channel. An attacker can leverage this by interposing between the communicating entities and masquerading as the original entity. In the absence of sufficient verification of identity, such an attacker can eavesdrop and potentially modify the communication between the original entities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

摘要确定了多种不同的可能性，表明这是一个必须分解为更具体的弱点的类别。

Weakness ID: 301

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Reflection Attack in an Authentication Protocol zh: -->身份验证协议中的反射攻击*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果恶意用户可以使用目标计算机模拟受信任的用户，则简单身份验证协议会受到反射攻击。

### Description:

Simple authentication protocols are subject to reflection attacks if a malicious user can use the target machine to impersonate a trusted user.

### 详细描述:

相互认证协议要求每一方通过用预共享密钥加密来响应另一方的随机质询。然而，这些协议通常使用相同的预共享密钥来与多个不同实体进行通信。恶意用户或攻击者可以通过对协议采用反射攻击来轻松地破坏此协议，而无需拥有正确的密钥。

### Extended Description:

A mutual authentication protocol requires each party to respond to a random challenge by the other party by encrypting it with a pre-shared key. Often, however, such protocols employ the same pre-shared key for communication with a number of different entities. A malicious user or an attacker can easily compromise this protocol without possessing the correct key by employing a reflection attack on the protocol.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

“反射”一词在CWE和社区中以多种方式使用，因此应该审查其用法。

Weakness ID: 302

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Authentication Bypass by Assumed-Immutable Data zh: -->认证绕过假定不可变数据*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

身份验证方案或实现使用假定为不可变的关键数据元素，但可由攻击者控制或修改。

### Description:

The authentication scheme or implementation uses key data elements that are assumed to be immutable, but can be controlled or modified by the attacker.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 303

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incorrect Implementation of Authentication Algorithm zh: -->验证算法的不正确实现*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件的要求决定了使用已建立的认证算法，但算法的实现是不正确的。

### Description:

The requirements for the software dictate the use of an established authentication algorithm, but the implementation of the algorithm is incorrect.

### 详细描述:

这种不正确的实现可能允许绕过身份验证。

### Extended Description:

This incorrect implementation may allow authentication to be bypassed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 304

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Missing Critical Step in Authentication zh: -->缺少身份验证的关键步骤*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件实现了一种身份验证技术，但它跳过了削弱技术的步骤。

### Description:

The software implements an authentication technique, but it skips a step that weakens the technique.

### 详细描述:

身份验证技术应遵循精确定义它们的算法，否则可以绕过身份验证或更容易受到暴力攻击。

### Extended Description:

Authentication techniques should follow the algorithms that define them exactly, otherwise authentication can be bypassed or more easily subjected to brute force attacks.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 305

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Authentication Bypass by Primary Weakness zh: -->主要弱点的身份验证旁路*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

验证算法是合理的，但是实现的机制可以被旁路，因为单独的弱点是认证错误的主要缺陷。

### Description:

The authentication algorithm is sound, but the implemented mechanism can be bypassed as the result of a separate weakness that is primary to the authentication error.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

大多数“身份验证绕过”错误是结果，而不是主要错误

Weakness ID: 306

提交日期 2010-02-16---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Missing Authentication for Critical Function zh: -->缺少关键功能的身份验证*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不对需要可证明的用户身份或消耗大量资源的功能执行任何身份验证。

### Description:

The software does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这与存在身份验证但存在错误的“绕过”问题是分开的。

Weakness ID: 307

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Restriction of Excessive Authentication Attempts zh: -->过度认证尝试的不当限制*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件没有实施足够的措施来防止在短时间内发生多次失败的身份验证尝试，使其更容易受到暴力攻击。

### Description:

The software does not implement sufficient measures to prevent multiple failed authentication attempts within in a short time frame, making it more susceptible to brute force attacks.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 308

提交日期 2008-04-11---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Use of Single-factor Authentication zh: -->使用单因素身份验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

与双因素身份验证方案的优势相比，使用单因素身份验证可能会导致不必要的危害风险。

### Description:

The use of single-factor authentication can lead to unnecessary risk of compromise when compared with the benefits of a dual-factor authentication scheme.

### 详细描述:

虽然使用多种身份验证方案只是在身份验证之上进行了更多的复杂化，但拥有这种冗​​余度量是非常有价值的。在互联网上使用弱，重用和通用密码是猖獗的。如果没有多个身份验证方案的额外保护，单个错误可能会导致帐户泄露。因此，如果可能有多种方案且易于使用，则应实施和要求它们。

### Extended Description:

While the use of multiple authentication schemes is simply piling on more complexity on top of authentication, it is inestimably valuable to have such measures of redundancy. The use of weak, reused, and common passwords is rampant on the internet. Without the added protection of multiple authentication schemes, a single mistake can result in the compromise of an account. For this reason, if multiple schemes are possible and also easy to use, they should be implemented and required.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 309

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of Password System for Primary Authentication zh: -->使用密码系统进行主要验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

使用密码系统作为主要的认证手段可能存在若干缺陷或缺点，每个缺陷或缺点都会降低机制的有效性。

### Description:

The use of password systems as the primary means of authentication may be subject to several flaws or shortcomings, each reducing the effectiveness of the mechanism.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

密码系统是最简单，最普遍的认证机制。然而，它们受到这种众所周知的攻击，并且经常受到损害，因为它们在最简单的实现中的使用是不实际的。

### 笔记 (Notes):

没有笔记

Weakness ID: 311

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Missing Encryption of Sensitive Data zh: -->缺少敏感数据的加密*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在存储或传输之前，该软件不会加密敏感或关键信息。

### Description:

The software does not encrypt sensitive or critical information before storage or transmission.

### 详细描述:

缺乏适当的数据加密传递了正确实施加密所传达的机密性，完整性和责任性的保证。

### Extended Description:

The lack of proper data encryption passes up the guarantees of confidentiality, integrity, and accountability that properly implemented encryption conveys.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

敏感信息的不安全存储（CWE-922）与敏感信息的缺失加密（CWE-311）之间存在重叠关系。加密通常用于防止攻击者读取敏感数据。但是，加密不会阻止攻击者擦除或覆盖数据。

Weakness ID: 312

提交日期 2009-01-12---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Cleartext Storage of Sensitive Information zh: -->明文存储敏感信息*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序将敏感信息以明文形式存储在可能可供另一个控制领域访问的资源中。

### Description:

The application stores sensitive information in cleartext within a resource that might be accessible to another control sphere.

### 详细描述:

由于信息以明文形式存储，因此攻击者可能会阅读它。即使以非人类可读的方式对信息进行编码，某些技术也可以确定正在使用哪种编码，然后对信息进行解码。

### Extended Description:

Because the information is stored in cleartext, attackers could potentially read it. Even if the information is encoded in a way that is not human-readable, certain techniques could determine which encoding is being used, then decode the information.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

不同的人使用“明文”和“明文”来表示同样的事情：缺乏加密。但是，在密码学中，这些具有更精确的含义。明文是将信息输入加密算法之前的信息，包括已加密的文本。 Cleartext是任何未加密的信息，尽管它可能是一种不易被人类阅读的编码形式（例如base64编码）。

Weakness ID: 313

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Cleartext Storage in a File or on Disk zh: -->文件或磁盘上的明文存储*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序将敏感信息以明文形式存储在文件或磁盘上。

### Description:

The application stores sensitive information in cleartext in a file, or on disk.

### 详细描述:

敏感信息可由具有文件访问权限的攻击者读取，或者通过对原始磁盘的物理或管理员访问权限读取。即使以非人类可读的方式对信息进行编码，某些技术也可以确定正在使用哪种编码，然后对信息进行解码。

### Extended Description:

The sensitive information could be read by attackers with access to the file, or with physical or administrator access to the raw disk. Even if the information is encoded in a way that is not human-readable, certain techniques could determine which encoding is being used, then decode the information.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

不同的人使用“明文”和“明文”来表示同样的事情：缺乏加密。但是，在密码学中，这些具有更精确的含义。明文是将信息输入加密算法之前的信息，包括已加密的文本。 Cleartext是任何未加密的信息，尽管它可能是一种不易被人类阅读的编码形式（例如base64编码）。

Weakness ID: 314

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Cleartext Storage in the Registry zh: -->注册表中的明文存储*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序将敏感信息以明文形式存储在注册表中。

### Description:

The application stores sensitive information in cleartext in the registry.

### 详细描述:

攻击者可以通过访问注册表项来读取信息。即使以非人类可读的方式对信息进行编码，某些技术也可以确定正在使用哪种编码，然后对信息进行解码。

### Extended Description:

Attackers can read the information by accessing the registry key. Even if the information is encoded in a way that is not human-readable, certain techniques could determine which encoding is being used, then decode the information.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

不同的人使用“明文”和“明文”来表示同样的事情：缺乏加密。但是，在密码学中，这些具有更精确的含义。明文是将信息输入加密算法之前的信息，包括已加密的文本。 Cleartext是任何未加密的信息，尽管它可能是一种不易被人类阅读的编码形式（例如base64编码）。

Weakness ID: 315

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Cleartext Storage of Sensitive Information in a Cookie zh: -->明文在Cookie中存储敏感信息*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序将敏感信息以明文形式存储在cookie中。

### Description:

The application stores sensitive information in cleartext in a cookie.

### 详细描述:

攻击者可以使用广泛使用的工具来查看cookie并读取敏感信息。即使以非人类可读的方式对信息进行编码，某些技术也可以确定正在使用哪种编码，然后对信息进行解码。

### Extended Description:

Attackers can use widely-available tools to view the cookie and read the sensitive information. Even if the information is encoded in a way that is not human-readable, certain techniques could determine which encoding is being used, then decode the information.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

不同的人使用“明文”和“明文”来表示同样的事情：缺乏加密。但是，在密码学中，这些具有更精确的含义。明文是将信息输入加密算法之前的信息，包括已加密的文本。 Cleartext是任何未加密的信息，尽管它可能是一种不易被人类阅读的编码形式（例如base64编码）。

Weakness ID: 316

提交日期 2013-07-17---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Cleartext Storage of Sensitive Information in Memory zh: -->明文中存储敏感信息的明文*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序将敏感信息以明文形式存储在内存中。

### Description:

The application stores sensitive information in cleartext in memory.

### 详细描述:

敏感内存可能会保存到磁盘，存储在核心转储中，或者如果应用程序崩溃仍未清除，或者程序员在释放内存之前未正确清除内存。  
可以说，这些问题通常只有具有管理员权限的人才可以利用。但是，交换可能会导致内存写入磁盘并使之后可以进行物理攻击。核心转储文件可能具有不安全的权限，或存储在不受信任的人可访问的归档文件中。或者，由于另一个弱点，未清除的敏感内存可能会无意中暴露给攻击者。

### Extended Description:

The sensitive memory might be saved to disk, stored in a core dump, or remain uncleared if the application crashes, or if the programmer does not properly clear the memory before freeing it.  
It could be argued that such problems are usually only exploitable by those with administrator privileges. However, swapping could cause the memory to be written to disk and leave it accessible to physical attack afterwards. Core dump files might have insecure permissions or be stored in archive files that are accessible to untrusted people. Or, uncleared sensitive memory might be inadvertently exposed to attackers due to another weakness.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能是一个结果性的弱点，例如如果编译器删除了旨在擦除内存的代码。

Weakness ID: 317

提交日期 2013-07-17---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Cleartext Storage of Sensitive Information in GUI zh: -->GUI中敏感信息的明文存储*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序在GUI中以明文形式存储敏感信息。

### Description:

The application stores sensitive information in cleartext within the GUI.

### 详细描述:

通过使用API​​直接访问GUI对象（如窗口和菜单），攻击者通常可以从GUI获取数据，即使是隐藏的。即使以非人类可读的方式对信息进行编码，某些技术也可以确定正在使用哪种编码，然后对信息进行解码。

### Extended Description:

An attacker can often obtain data from a GUI, even if hidden, by using an API to directly access GUI objects such as windows and menus. Even if the information is encoded in a way that is not human-readable, certain techniques could determine which encoding is being used, then decode the information.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

不同的人使用“明文”和“明文”来表示同样的事情：缺乏加密。但是，在密码学中，这些具有更精确的含义。明文是将信息输入加密算法之前的信息，包括已加密的文本。 Cleartext是任何未加密的信息，尽管它可能是一种不易被人类阅读的编码形式（例如base64编码）。

Weakness ID: 318

提交日期 2013-07-17---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Cleartext Storage of Sensitive Information in Executable zh: -->在可执行文件中明确存储敏感信息*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序将敏感信息以明文形式存储在可执行文件中。

### Description:

The application stores sensitive information in cleartext in an executable.

### 详细描述:

攻击者可以对二进制代码进行逆向工程以获取秘密数据当明文是纯ASCII时，这一点尤其容易。即使以非人类可读的方式对信息进行编码，某些技术也可以确定正在使用哪种编码，然后对信息进行解码。

### Extended Description:

Attackers can reverse engineer binary code to obtain secret data. This is especially easy when the cleartext is plain ASCII. Even if the information is encoded in a way that is not human-readable, certain techniques could determine which encoding is being used, then decode the information.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

不同的人使用“明文”和“明文”来表示同样的事情：缺乏加密。但是，在密码学中，这些具有更精确的含义。明文是将信息输入加密算法之前的信息，包括已加密的文本。 Cleartext是任何未加密的信息，尽管它可能是一种不易被人类阅读的编码形式（例如base64编码）。

Weakness ID: 319

提交日期 2009-01-12---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Cleartext Transmission of Sensitive Information zh: -->明文传播敏感信息*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件以通信渠道的明文形式传输敏感或安全关键数据，这些数据可能被未经授权的参与者嗅探。

### Description:

The software transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors.

### 详细描述:

在数据传输期间，攻击者可以“嗅探”许多通信信道。例如，任何有权访问网络接口的攻击者都可以嗅探网络流量。这大大降低了攻击者利用的难度。

### Extended Description:

Many communication channels can be "sniffed" by attackers during data transmission. For example, network traffic can often be sniffed by any attacker who has access to a network interface. This significantly lowers the difficulty of exploitation by attackers.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 321

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of Hard-coded Cryptographic Key zh: -->使用硬编码密码密钥*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

使用硬编码的加密密钥显着增加了可以恢复加密数据的可能性。

### Description:

The use of a hard-coded cryptographic key significantly increases the possibility that encrypted data may be recovered.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

使用硬编码密码和使用硬编码加密密钥之间的主要区别在于前者传达的错误安全感。许多人认为，在存储之前简单地散列硬编码密码将保护信息免受恶意用户的攻击。但是，许多哈希值是可逆的（或者至少容易受到暴力攻击） - 而且，许多身份验证协议只是简单地请求哈希本身，这使得它不比密码更好。

Weakness ID: 322

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Key Exchange without Entity Authentication zh: -->没有实体身份验证的密钥交换*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件与演员进行密钥交换，而不验证该演员的身份。

### Description:

The software performs a key exchange with an actor without verifying the identity of that actor.

### 详细描述:

执行密钥交换将保持两个实体之间发送的信息的完整性，但这并不能保证实体是他们声称的实体。这可能会导致一系列“中间人”攻击。通常，这涉及受害客户端，该客户端联系冒充受信任服务器的恶意服务器。如果客户端跳过身份验证或忽略身份验证失败，则恶意服务器可以从用户请求身份验证信息。然后，恶意服务器可以使用此身份验证信息使用受害者的凭据登录受信任的服务器，嗅探受害者和受信任服务器之间的流量等。

### Extended Description:

Performing a key exchange will preserve the integrity of the information sent between two entities, but this will not guarantee that the entities are who they claim they are. This may enable a set of "man-in-the-middle" attacks. Typically, this involves a victim client that contacts a malicious server that is impersonating a trusted server. If the client skips authentication or ignores an authentication failure, the malicious server may request authentication information from the user. The malicious server can then use this authentication information to log in to the trusted server using the victim's credentials, sniff traffic between the victim and trusted server, etc.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 323

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Reusing a Nonce, Key Pair in Encryption zh: -->在加密中重用Nonce，密钥对*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Nonce应该用于当前场合并且只能使用一次。

### Description:

Nonces should be used for the present occasion and only once.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

随机数通常与通信交换中的密钥捆绑在一起，以便为每个交换产生新的会话密钥。

### 笔记 (Notes):

没有笔记

Weakness ID: 324

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of a Key Past its Expiration Date zh: -->在过期日期之后使用密钥*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用加密密钥或密码超过其到期日期，通过增加针对该密钥的破解攻击的时间窗口，显着降低了其安全性。

### Description:

The product uses a cryptographic key or password past its expiration date, which diminishes its safety significantly by increasing the timing window for cracking attacks against that key.

### 详细描述:

虽然密钥的到期并不一定确保它们被泄露，但是长时间使用的密钥具有降低的完整性概率是一个重要的问题。因此，在与其强度成比例的时间段内更换按键非常重要。

### Extended Description:

While the expiration of keys does not necessarily ensure that they are compromised, it is a significant concern that keys which remain in use for prolonged periods of time have a decreasing probability of integrity. For this reason, it is important to replace keys within a period of time proportional to their strength.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 325

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Missing Required Cryptographic Step zh: -->缺少必需的加密步骤*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件没有在加密算法中实现所需的步骤，导致加密比该算法所公布的更弱。

### Description:

The software does not implement a required step in a cryptographic algorithm, resulting in weaker encryption than advertised by that algorithm.

### 详细描述:

加密实现应遵循精确定义它们的算法，否则加密可能比预期的要弱。

### Extended Description:

Cryptographic implementations should follow the algorithms that define them exactly, otherwise encryption can be weaker than expected.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠不完整/缺失的安全检查。

Weakness ID: 326

提交日期 2009-07-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Inadequate Encryption Strength zh: -->加密强度不足*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用理论上合理的加密方案存储或传输敏感数据，但不足以达到所需的保护级别。

### Description:

The software stores or transmits sensitive data using an encryption scheme that is theoretically sound, but is not strong enough for the level of protection required.

### 详细描述:

弱加密方案可能会遭受暴力攻击，这些攻击有可能成功使用当前的攻击方法和资源。

### Extended Description:

A weak encryption scheme can be subjected to brute force attacks that have a reasonable chance of succeeding using current attack methods and resources.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

存在各种加密算法，具有各种弱点。此类别可能会拆分为较小的子类别。

Weakness ID: 327

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of a Broken or Risky Cryptographic Algorithm zh: -->使用破碎或危险的密码算法*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

使用破坏或有风险的加密算法是不必要的风险，可能导致敏感信息的暴露。

### Description:

The use of a broken or risky cryptographic algorithm is an unnecessary risk that may result in the exposure of sensitive information.

### 详细描述:

使用非标准算法是危险的，因为确定的攻击者可能能够破坏算法并破坏任何受保护的数据。可能存在众所周知的技术来破坏算法。

### Extended Description:

The use of a non-standard algorithm is dangerous because a determined attacker may be able to break the algorithm and compromise whatever data has been protected. Well-known techniques may exist to break the algorithm.

### 问题背景 (Background Detail):

加密算法是数据加扰的方法。大多数应用程序都应该使用少量易于理解且经过深入研究的算法。生成安全算法非常困难，甚至已经完成的加密专家的高调算法也被打破了。由于密码学的进展如此迅速，因此即使曾经认为算法很强，算法也被认为是“不安全的”。当发现针对算法的新攻击时，或者如果计算能力增加太多以至于加密算法不再提供最初认为的保护量时，就会发生这种情况。

### 笔记 (Notes):

需要对CWE-310，CWE-326和CWE-327与其所有子女之间的关系进行审查和重组。

Weakness ID: 328

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Reversible One-Way Hash zh: -->可逆的单向哈希*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用散列算法生成哈希值，该哈希值可用于确定原始输入，或查找可生成相同哈希的输入，比蛮力技术更有效。

### Description:

The product uses a hashing algorithm that produces a hash value that can be used to determine the original input, or to find an input that can produce the same hash, more efficiently than brute force techniques.

### 详细描述:

当哈希用于需要单向属性保持的安全算法时，这种弱点尤其危险。例如，如果身份验证系统获取传入密码并生成哈希值，则将哈希值与其身份验证数据库中存储的另一个哈希值进行比较，然后创建冲突的能力可能允许攻击者提供生成的备用密码相同的目标哈希，绕过身份验证。

### Extended Description:

This weakness is especially dangerous when the hash is used in security algorithms that require the one-way property to hold. For example, if an authentication system takes an incoming password and generates a hash, then compares the hash to another hash that it has stored in its authentication database, then the ability to create a collision could allow an attacker to provide an alternate password that produces the same target hash, bypassing authentication.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 329

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Not Using a Random IV with CBC Mode zh: -->不使用具有CBC模式的随机IV*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

不使用具有密码块链接（CBC）模式的随机初始化向量（IV）导致算法易受字典攻击。

### Description:

Not using a random initialization Vector (IV) with Cipher Block Chaining (CBC) Mode causes algorithms to be susceptible to dictionary attacks.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

CBC是分组密码最常用的操作模式。它通过用明文对密文进行异或来解决电子代码簿的字典问题。如果它用于加密多个数据流，则字典攻击是可能的，只要这些流具有共同的开始序列。

### 笔记 (Notes):

没有笔记

Weakness ID: 330

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Insufficiently Random Values zh: -->使用不充分的随机值*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Usable*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件可能在安全上下文中使用不完全随机的数字或值，这取决于不可预测的数字。

### Description:

The software may use insufficiently random numbers or values in a security context that depends on unpredictable numbers.

### 详细描述:

当软件在需要不可预测性的上下文中生成可预测值时，攻击者可能猜测将生成的下一个值，并使用此猜测来模拟其他用户或访问敏感信息。

### Extended Description:

When software generates predictable values in a context requiring unpredictability, it may be possible for an attacker to guess the next value that will be generated, and use this guess to impersonate another user or access sensitive information.

### 问题背景 (Background Detail):

计算机是确定性机器，因此无法产生真正的随机性。伪随机数发生器（PRNG）在算法上近似随机性，从计算后续值的种子开始。有两种类型的PRNG：统计和加密。统计PRNG提供有用的统计属性，但它们的输出是高度可预测的，并且形成易于重现的数字流，不适合在安全性取决于生成的值不可预测的情况下使用。加密PRNG通过生成更难以预测的输出来解决此问题。对于密码安全的值，攻击者必须不可能或极不可能区分它和真正的随机值。

### 笔记 (Notes):

这可能是许多其他弱点的主要原因，例如加密错误，身份验证错误，符号链接跟踪，信息泄漏等。

Weakness ID: 331

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Insufficient Entropy zh: -->熵不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用产生不足熵的算法或方案，留下比其他更可能发生的模式或值集群。

### Description:

The software uses an algorithm or scheme that produces insufficient entropy, leaving patterns or clusters of values that are more likely to occur than others.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 332

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Insufficient Entropy in PRNG zh: -->PRNG中的熵不足*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

伪随机数发生器（PRNG）可用或使用的熵的缺乏可能是稳定性和安全性威胁。

### Description:

The lack of entropy available for, or used by, a Pseudo-Random Number Generator (PRNG) can be a stability and security threat.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 333

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Handling of Insufficient Entropy in TRNG zh: -->TRNG中熵不足的处理不当*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

真随机数发生器（TRNG）通常具有有限的熵源，因此可能失败或阻塞。

### Description:

True random number generators (TRNG) generally have a limited source of entropy and therefore can fail or block.

### 详细描述:

可以生成真随机数的速率是有限的。重要的是，只有在安全需要时才使用它们。

### Extended Description:

The rate at which true random numbers can be generated is limited. It is important that one uses them only when they are needed for security.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 334

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Small Space of Random Values zh: -->随机值的小空间*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

可能的随机值的数量小于产品所需的数量，使其更容易受到强力攻击。

### Description:

The number of possible random values is smaller than needed by the product, making it more susceptible to brute force attacks.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 335

提交日期 2017-11-08---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incorrect Usage of Seeds in Pseudo-Random Number Generator (PRNG) zh: -->伪随机数发生器（PRNG）中种子的使用不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用伪随机数发生器（PRNG），不能正确管理种子。

### Description:

The software uses a Pseudo-Random Number Generator (PRNG) that does not correctly manage seeds.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 336

提交日期 2017-11-08---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Same Seed in Pseudo-Random Number Generator (PRNG) zh: -->伪随机数发生器（PRNG）中的相同种子*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

伪随机数发生器（PRNG）在每次初始化产品时使用相同的种子。

### Description:

A Pseudo-Random Number Generator (PRNG) uses the same seed each time the product is initialized.

### 详细描述:

如果攻击者可以猜测（或知道）种子，则攻击者可能能够确定将从PRNG生成的随机数。

### Extended Description:

If an attacker can guess (or knows) the seed, then the attacker may be able to determine the random numbers that will be produced from the PRNG.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 337

提交日期 2017-11-08---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Predictable Seed in Pseudo-Random Number Generator (PRNG) zh: -->伪随机数发生器（PRNG）中的可预测种子*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

伪随机数发生器（PRNG）从可预测的种子初始化，例如进程ID或系统时间。

### Description:

A Pseudo-Random Number Generator (PRNG) is initialized from a predictable seed, such as the process ID or system time.

### 详细描述:

可预测种子的使用显着减少了攻击者需要测试的可能种子的数量，以便预测PRNG将生成哪些随机numnbers。

### Extended Description:

The use of predictable seeds significantly reduces the number of possible seeds that an attacker would need to test in order to predict which random numnbers will be generated by the PRNG.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 338

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG) zh: -->使用密码弱伪随机数发生器（PRNG）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品在安全上下文中使用伪随机数生成器（PRNG），但PRNG的算法在加密方面不强。

### Description:

The product uses a Pseudo-Random Number Generator (PRNG) in a security context, but the PRNG's algorithm is not cryptographically strong.

### 详细描述:

当在加密上下文中使用非加密PRNG时，它可以将加密暴露给某些类型的攻击。  
通常，伪随机数发生器（PRNG）不是为加密而设计的。对于使用随机数的算法，有时候平庸的随机性来源是充分的或者更可取的。弱发电机通常需要较少的处理能力和/或不在系统上使用宝贵的有限熵源。虽然这些PRNG可能具有非常有用的功能，但这些相同的功能可用于打破加密。

### Extended Description:

When a non-cryptographic PRNG is used in a cryptographic context, it can expose the cryptography to certain types of attacks.  
Often a pseudo-random number generator (PRNG) is not designed for cryptography. Sometimes a mediocre source of randomness is sufficient or preferable for algorithms that use random numbers. Weak generators generally take less processing power and/or do not use the precious, finite, entropy sources on a system. While such PRNGs might have very useful features, these same features could be used to break the cryptography.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 339

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Small Seed Space in PRNG zh: -->PRNG的小种子空间*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

PRNG使用相对较小的种子空间。

### Description:

A PRNG uses a relatively small space of seeds.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目与可观察状态（CWE-341）的可预测重叠。

Weakness ID: 340

提交日期 I`m don`t know---> 修改日期 2012-05-11

* **Weakness Name:** *en: --> Predictability Problems zh: -->可预测性问题*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此类别中的弱点与生成数字或标识符的方案相关，这些数字或标识符比应用程序所需的更可预测。

### Description:

Weaknesses in this category are related to schemes that generate numbers or identifiers that are more predictable than required by the application.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 341

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Predictable from Observable State zh: -->从可观察国家可预测*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

根据攻击者可以对系统或网络的状态（例如时间，进程ID等）进行的观察，可以预测数字或对象。

### Description:

A number or object is predictable based on observations that the attacker can make about the state of the system or network, such as time, process ID, etc.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 342

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Predictable Exact Value from Previous Values zh: -->先前值的可预测精确值*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

通过观察先前的值可以精确地预测精确值或随机数。

### Description:

An exact value or random number can be precisely predicted by observing previous values.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 343

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Predictable Value Range from Previous Values zh: -->先前值的可预测值范围*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件的随机数发生器产生一系列值，当观察到这些值时，可用于推断可能产生的下一个值的相对小范围的可能性。

### Description:

The software's random number generator produces a series of values which, when observed, can be used to infer a relatively small range of possibilities for the next value that could be generated.

### 详细描述:

根据对先前值的观察，不应预测随机数发生器的输出。在某些情况下，攻击者无法预测接下来会产生的确切值，但可以显着缩小可能性。这减少了执行暴力攻击的努力量。例如，假设产品生成1到100之间的随机数，但它总是产生一个更大的值，直到达到100.如果生成器产生80，那么攻击者知道下一个值将介于81和100之间。在100种可能性中，攻击者只需要考虑20种。

### Extended Description:

The output of a random number generator should not be predictable based on observations of previous values. In some cases, an attacker cannot predict the exact value that will be produced next, but can narrow down the possibilities significantly. This reduces the amount of effort to perform a brute force attack. For example, suppose the product generates random numbers between 1 and 100, but it always produces a larger value until it reaches 100. If the generator produces an 80, then the attacker knows that the next value will be somewhere between 81 and 100. Instead of 100 possibilities, the attacker only needs to consider 20.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 344

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of Invariant Value in Dynamically Changing Context zh: -->在动态变化的上下文中使用不变值*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用常量值，名称或引用，但此值可以（或应该）在不同的环境中变化。

### Description:

The product uses a constant value, name, or reference, but this value can (or should) vary across different environments.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠默认配置。

Weakness ID: 345

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Insufficient Verification of Data Authenticity zh: -->数据真实性验证不足*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法充分验证数据的来源或真实性，导致其接受无效数据。

### Description:

The software does not sufficiently verify the origin or authenticity of data, in a way that causes it to accept invalid data.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

“原产地验证”可能属于这一点。

Weakness ID: 346

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Origin Validation Error zh: -->原点验证错误*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件未正确验证数据源或通信是否有效。

### Description:

The software does not properly verify that the source of data or communication is valid.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目与其他CWE条目有一些重要的重叠，可能需要一些说明。参见术语说明。

Weakness ID: 347

提交日期 2009-05-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Verification of Cryptographic Signature zh: -->密码签名的不正确验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会验证或错误地验证数据的加密签名。

### Description:

The software does not verify, or incorrectly verifies, the cryptographic signature for data.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 348

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of Less Trusted Source zh: -->使用较少可信源*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件具有相同数据或信息的两个不同来源，但它使用的源代码对验证的支持较少，信任度较低，或者抵御攻击的能力较弱。

### Description:

The software has two different sources of the same data or information, but it uses the source that has less support for verification, is less trusted, or is less resistant to attack.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 349

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Acceptance of Extraneous Untrusted Data With Trusted Data zh: -->使用可信数据接受外部不受信任的数据*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在处理可信数据时，接受可信数据中包含的任何不受信任的数据，将不受信任的数据视为可信数据。

### Description:

The software, when processing trusted data, accepts any untrusted data that is also included with the trusted data, treating the untrusted data as if it were trusted.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 350

提交日期 2013-07-17---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Reliance on Reverse DNS Resolution for a Security-Critical Action zh: -->依赖反向DNS解决方案来实现安全关键行动*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件对IP地址执行反向DNS解析以获取主机名并做出安全决策，但它无法正确确保IP地址与主机名真正关联。

### Description:

The software performs reverse DNS resolution on an IP address to obtain the hostname and make a security decision, but it does not properly ensure that the IP address is truly associated with the hostname.

### 详细描述:

由于DNS名称很容易被欺骗或误报，并且软件可能难以检测可信DNS服务器是否已被泄露，因此DNS名称不构成有效的认证机制。  
当软件对IP地址执行反向DNS解析时，如果攻击者控制服务器获取该IP地址，则攻击者可以使服务器返回任意主机名。因此，攻击者可能绕过身份验证，导致错误的主机名记录在日志文件中以隐藏活动或执行其他攻击。  
攻击者可以通过（1）破坏DNS服务器并修改其记录（有时称为DNS缓存中毒）或（2）对与其IP地址关联的DNS服务器进行合法控制来欺骗DNS名称。

### Extended Description:

Since DNS names can be easily spoofed or misreported, and it may be difficult for the software to detect if a trusted DNS server has been compromised, DNS names do not constitute a valid authentication mechanism.  
When the software performs a reverse DNS resolution for an IP address, if an attacker controls the server for that IP address, then the attacker can cause the server to return an arbitrary hostname. As a result, the attacker may be able to bypass authentication, cause the wrong hostname to be recorded in log files to hide activities, or perform other attacks.  
Attackers can spoof DNS names by either (1) compromising a DNS server and modifying its records (sometimes called DNS cache poisoning), or (2) having legitimate control over a DNS server associated with their IP address.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

CWE-350，CWE-247和CWE-292在CWE 2.5中合并为CWE-350。 CWE-247最初来自七个恶性王国，来自PLOVER的CWE-350和来自CLASP的CWE-292。所有分类法都密切关注使用反向DNS来验证传入请求。

Weakness ID: 351

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Insufficient Type Distinction zh: -->类型区别不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法以导致不安全行为的方式正确区分不同类型的元素。

### Description:

The software does not properly distinguish between different types of elements in a way that leads to insecure behavior.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠其他人，例如多重解释错误。

Weakness ID: 352

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Cross-Site Request Forgery (CSRF) zh: -->跨站请求伪造（CSRF）*
* **Abstraction:** *Compound* **Structure:** *Composite* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序没有或不能充分验证提交请求的用户是否有意提供了格式良好，有效，一致的请求。

### Description:

The web application does not, or can not, sufficiently verify whether a well-formed, valid, consistent request was intentionally provided by the user who submitted the request.

### 详细描述:

当Web服务器被设计为从客户端接收请求而没有任何机制来验证它是否被故意发送时，攻击者可能会欺骗客户端向Web服务器发出无意的请求，该请求将被视为真实的要求。这可以通过URL，图像加载，XMLHttpRequest等来完成，并且可能导致数据暴露或意外的代码执行。

### Extended Description:

When a web server is designed to receive a request from a client without any mechanism for verifying that it was intentionally sent, then it might be possible for an attacker to trick a client into making an unintentional request to the web server which will be treated as an authentic request. This can be done via a URL, image load, XMLHttpRequest, etc. and can result in exposure of data or unintended code execution.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

XSS和CSRF（CWE-352）之间可能存在密切关系。攻击者可能使用CSRF来诱骗受害者向请求包含XSS有效负载的服务器提交请求。一个众所周知的例子就是MySpace上的Samy蠕虫[REF-956]。蠕虫使用XSS将恶意HTML序列插入用户的个人资料中，并将攻击者添加为MySpace朋友。然后，该受害者的MySpace朋友将执行有效负载以修改他们自己的配置文件，从而导致蠕虫以指数方式传播。由于受害者本身并未故意插入恶意脚本，因此CSRF是一个根本原因。

Weakness ID: 353

提交日期 2010-12-13---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Missing Support for Integrity Check zh: -->缺少对完整性检查的支持*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用传输协议，该协议不包括用于在传输期间验证数据完整性的机制，例如校验和。

### Description:

The software uses a transmission protocol that does not include a mechanism for verifying the integrity of the data during transmission, such as a checksum.

### 详细描述:

如果从协议中省略完整性检查值或“校验和”，则无法确定数据在传输中是否已损坏。协议中缺少校验和功能会删除可以使用的数据的第一个应用程序级别检查。端到端的检查理念指出，完整性检查应该在可以完全实现的最低级别执行。除了应用程序执行的进一步的健全性检查和输入验证之外，协议的校验和是最重要的校验和级别，因为它可以比任何先前级别更完整地执行并且考虑整个消息，而不是单个数据包。

### Extended Description:

If integrity check values or "checksums" are omitted from a protocol, there is no way of determining if data has been corrupted in transmission. The lack of checksum functionality in a protocol removes the first application-level check of data that can be used. The end-to-end philosophy of checks states that integrity checks should be performed at the lowest level that they can be completely implemented. Excluding further sanity checks and input validation performed by applications, the protocol's checksum is the most important level of checksum, since it can be performed more completely than at any previous level and takes into account entire messages, as opposed to single packets.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 354

提交日期 2009-03-10---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Validation of Integrity Check Value zh: -->对完整性检查值的不正确验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不验证或错误地验证消息的完整性检查值或“校验和”。这可以防止它检测数据是否在传输中被修改或损坏。

### Description:

The software does not validate or incorrectly validates the integrity check values or "checksums" of a message. This may prevent it from detecting if the data has been modified or corrupted in transmission.

### 详细描述:

使用前对校验和的不正确验证会导致不必要的风险，可以轻松减轻。协议规范描述了用于计算校验和的算法。然后，实现计算并验证计算的校验和与接收的校验和匹配是一件简单的事情。对计算的校验和和接收的校验和的不正确验证可能导致更大的后果。

### Extended Description:

Improper validation of checksums before use results in an unnecessary risk that can easily be mitigated. The protocol specification describes the algorithm used for calculating the checksum. It is then a simple matter of implementing the calculation and verifying that the calculated checksum and the received checksum match. Improper verification of the calculated checksum and the received checksum can lead to far greater consequences.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 356

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Product UI does not Warn User of Unsafe Actions zh: -->产品UI不会警告不安全操作的用户*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在代表该用户执行不安全操作之前，该软件的用户界面不会警告用户。这使攻击者更容易诱骗用户对其系统造成损害。

### Description:

The software's user interface does not warn the user before undertaking an unsafe action on behalf of that user. This makes it easier for attackers to trick users into inflicting damage to their system.

### 详细描述:

软件系统应警告用户，如果用户继续进行，可能会发生潜在危险的操作。例如，如果用户从未知来源下载文件并尝试在其计算机上执行该文件，则应用程序的GUI可以指示该文件不安全。

### Extended Description:

Software systems should warn users that a potentially dangerous action may occur if the user proceeds. For example, if the user downloads a file from an unknown source and attempts to execute the file on their machine, then the application's GUI can indicate that the file is unsafe.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

经常产生，例如在未处理的错误条件下。

Weakness ID: 357

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Insufficient UI Warning of Dangerous Operations zh: -->危险操作的UI警告不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

用户界面向用户提供有关危险或敏感操作的警告，但警告不足以引起注意。

### Description:

The user interface provides a warning to a user regarding dangerous or sensitive operations, but the warning is not noticeable enough to warrant attention.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 358

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improperly Implemented Security Check for Standard zh: -->未正确实施标准安全检查*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件未实现或错误地实现由标准化算法，协议或技术的设计指定的一个或多个安全相关检查。

### Description:

The software does not implement or incorrectly implements one or more security-relevant checks as specified by the design of a standardized algorithm, protocol, or technique.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这是产品方面的“缺失步骤”错误，可能会与验证不足和欺骗等缺陷重叠。它经常出现在加密和身份验证错误中。有时会产生这种结果。

Weakness ID: 359

提交日期 2014-02-18---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Exposure of Private Information ('Privacy Violation') zh: -->私人信息的曝光（'隐私违规'）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法正确防止私人数据（如信用卡号）被（1）未明确授权访问数据或（2）未获得数据所针对的人的默许相关的。

### Description:

The software does not properly prevent private data (such as credit card numbers) from being accessed by actors who either (1) are not explicitly authorized to access the data or (2) do not have the implicit consent of the people to which the data is related.

### 详细描述:

错误处理私人信息（例如客户密码或社会安全号码）可能会损害用户隐私，而且通常是非法的。私人信息的曝光并不一定会妨碍软件正常工作，事实上它可能是开发人员想要的，但对于与此私人信息相关的人来说，它仍然是不受欢迎的（或法律明确禁止）。  
在以下情况下可能发生隐私  
  
  
私人用户信息进入该程序。  
数据将写入外部位置，例如控制台，文件系统或网络。  
  
  
私人数据可以通过多种方式进入程序：  
  
  
直接来自用户的密码或个人信息的形式  
应用程序从数据库或其他数据存储访问  
间接来自合作伙伴或其他第三方  
  
  
某些类型的私人信息包括：  
  
  
政府标识符，例如社会安全号码  
联系信息，如家庭住址和电话号码  
地理位置 - 用户所在的位置  
工作经历  
财务数据 - 例如信用卡号，工资，银行账户和债务  
图片，视频或音频  
行为模式 - 例如网上冲浪历史，执行某些活动时等。  
与他人的关系（和关系类型） - 家人，朋友，联系人等。  
通讯 - 电子邮件地址，私人电子邮件，短信，聊天记录等。  
健康 - 医疗条件，保险状况，处方记录  
凭证，例如密码，可用于访问其他信息。  
  
  
这些信息中的一些可以被表征为PII（个人可识别信息），受保护的健康信息（PHI）等。私人信息的类别可以基于特定行业的预期用途或政策和实践而重叠或变化。  
根据其所在地，所经营的业务类型以及其处理的任何私人数据的性质，组织可能需要遵守以下一项或多项联邦和州法规： - 安全港隐私框架[REF-340 ] - Gramm-Leach Bliley Act（GLBA）[REF-341] - 健康保险流通与责任法案（HIPAA）[REF-342] - 加州SB-1386 [REF-343]。  
有时，未标记为私有的数据可能在不同的上下文中具有隐私含义。例如，学生识别号码通常不被视为私人，因为没有明确且公开可用的映射到个别学生的个人信息。但是，如果学校根据学生的社会安全号码生成识别号码，则识别号码应视为私密号码。  
安全和隐私问题似乎经常相互竞争。从安全角度来看，应记录所有重要操作，以便以后可以识别任何异常活动。但是，当涉及私人数据时，这种做法实际上可能会产生风险。尽管有许多方法可以不安全地处理私人数据，但共同的风险源于错误的信任。程序员通常信任程序运行的操作环境，因此认为可以接受在文件系统，注册表或其他本地控制的资源中存储私有信息。但是，即使限制对某些资源的访问，也不能保证可以信任具有访问权限的个人。

### Extended Description:

Mishandling private information, such as customer passwords or Social Security numbers, can compromise user privacy and is often illegal. An exposure of private information does not necessarily prevent the software from working properly, and in fact it might be intended by the developer, but it can still be undesirable (or explicitly prohibited by law) for the people who are associated with this private information.  
Privacy violations may occur when:  
  
  
Private user information enters the program.  
The data is written to an external location, such as the console, file system, or network.  
  
  
Private data can enter a program in a variety of ways:  
  
  
Directly from the user in the form of a password or personal information  
Accessed from a database or other data store by the application  
Indirectly from a partner or other third party  
  
  
Some types of private information include:  
  
  
Government identifiers, such as Social Security Numbers  
Contact information, such as home addresses and telephone numbers  
Geographic location - where the user is (or was)  
Employment history  
Financial data - such as credit card numbers, salary, bank accounts, and debts  
Pictures, video, or audio  
Behavioral patterns - such as web surfing history, when certain activities are performed, etc.  
Relationships (and types of relationships) with others - family, friends, contacts, etc.  
Communications - e-mail addresses, private e-mail messages, SMS text messages, chat logs, etc.  
Health - medical conditions, insurance status, prescription records  
Credentials, such as passwords, which can be used to access other information.  
  
  
Some of this information may be characterized as PII (Personally Identifiable Information), Protected Health Information (PHI), etc. Categories of private information may overlap or vary based on the intended usage or the policies and practices of a particular industry.  
Depending on its location, the type of business it conducts, and the nature of any private data it handles, an organization may be required to comply with one or more of the following federal and state regulations: - Safe Harbor Privacy Framework [REF-340] - Gramm-Leach Bliley Act (GLBA) [REF-341] - Health Insurance Portability and Accountability Act (HIPAA) [REF-342] - California SB-1386 [REF-343].  
Sometimes data that is not labeled as private can have a privacy implication in a different context. For example, student identification numbers are usually not considered private because there is no explicit and publicly-available mapping to an individual student's personal information. However, if a school generates identification numbers based on student social security numbers, then the identification numbers should be considered private.  
Security and privacy concerns often seem to compete with each other. From a security perspective, all important operations should be recorded so that any anomalous activity can later be identified. However, when private data is involved, this practice can in fact create risk. Although there are many ways in which private data can be handled unsafely, a common risk stems from misplaced trust. Programmers often trust the operating environment in which a program runs, and therefore believe that it is acceptable store private information on the file system, in the registry, or in other locally-controlled resources. However, even if access to certain resources is restricted, this does not guarantee that the individuals who do have access can be trusted.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 360

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Trust of System Event Data zh: -->信任系统事件数据*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

基于事件位置的安全性是不安全的，可能是欺骗性的。

### Description:

Security based on event locations are insecure and can be spoofed.

### 详细描述:

事件是消息传递系统，其可以向监听事件的程序提供控制数据。事件通常没有任何类型的身份验证框架，以允许从受信任的来源验证它们。 Windows中的任何应用程序都可以在给定桌面上向同一桌面上的任何窗口发送消息。这些消息没有身份验证框架。因此，如果进程未检查这些消息的有效性和安全性，则可以使用任何消息来操作桌面上的任何进程。

### Extended Description:

Events are a messaging system which may provide control data to programs listening for events. Events often do not have any type of authentication framework to allow them to be verified from a trusted source. Any application, in Windows, on a given desktop can send a message to any window on the same desktop. There is no authentication framework for these messages. Therefore, any message can be used to manipulate any process on the desktop if the process does not check the validity and safeness of those messages.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 362

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') zh: -->使用具有不正确同步的共享资源并发执行（'竞争条件'）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序包含可与其他代码并发运行的代码序列，并且代码序列需要对共享资源的临时，独占访问，但存在时序窗口，其中共享资源可由另一个并发操作的代码序列修改。

### Description:

The program contains a code sequence that can run concurrently with other code, and the code sequence requires temporary, exclusive access to a shared resource, but a timing window exists in which the shared resource can be modified by another code sequence that is operating concurrently.

### 详细描述:

当预期的同步在安全关键代码中时，这可能具有安全隐患，例如记录用户是否经过身份验证或修改不应受到局外人影响的重要状态信息。  
竞争条件发生在并发环境中，并且实际上是代码序列的属性。根据上下文，代码序列可以是函数调用，少量指令，一系列程序调用等形式。  
竞争条件违反了这些与之密切相关的属性：  
  
  
排他性 - 给予代码序列对共享资源的独占访问权，即，在原始序列完成执行之前，没有其他代码序列可以修改共享资源的属性。  
原子性 - 代码序列是行为原子的，即，没有其他线程或进程可以同时针对相同资源执行相同的指令序列（或子集）。  
  
  
当“干扰代码序列”仍然可以访问共享资源时，存在竞争条件，违反了排他性。程序员可以假设某些代码序列执行得太快而不受干扰代码序列的影响;如果不是，那就违反了原子性。例如，单个“x ++”语句在代码层可能看起来是原子的，但它在指令层实际上是非原子的，因为它涉及读取（x的原始值），然后是计算（x + 1） ），然后写入（将结果保存到x）。  
干扰代码序列可以是“可信任的”或“不可信的”。可信的干扰码序列发生在程序内;它不能被攻击者修改，只能间接调用。不受信任的干扰代码序列可以由攻击者直接创作，通常它位于易受攻击的程序外部。

### Extended Description:

This can have security implications when the expected synchronization is in security-critical code, such as recording whether a user is authenticated or modifying important state information that should not be influenced by an outsider.  
A race condition occurs within concurrent environments, and is effectively a property of a code sequence. Depending on the context, a code sequence may be in the form of a function call, a small number of instructions, a series of program invocations, etc.  
A race condition violates these properties, which are closely related:  
  
  
Exclusivity - the code sequence is given exclusive access to the shared resource, i.e., no other code sequence can modify properties of the shared resource before the original sequence has completed execution.  
Atomicity - the code sequence is behaviorally atomic, i.e., no other thread or process can concurrently execute the same sequence of instructions (or a subset) against the same resource.  
  
  
A race condition exists when an "interfering code sequence" can still access the shared resource, violating exclusivity. Programmers may assume that certain code sequences execute too quickly to be affected by an interfering code sequence; when they are not, this violates atomicity. For example, the single "x++" statement may appear atomic at the code layer, but it is actually non-atomic at the instruction layer, since it involves a read (the original value of x), followed by a computation (x+1), followed by a write (save the result to x).  
The interfering code sequence could be "trusted" or "untrusted." A trusted interfering code sequence occurs within the program; it cannot be modified by the attacker, and it can only be invoked indirectly. An untrusted interfering code sequence can be authored directly by the attacker, and typically it is external to the vulnerable program.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

需要进一步发展竞争条件和同步问题（CWE-662）之间的关系。它们不一定是同一核心概念的两个视角，因为同步只是避免竞争条件的一种技术，并且除了竞争条件预防之外，同步可以用于其他目的。

Weakness ID: 363

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Race Condition Enabling Link Following zh: -->竞争条件启用链接跟随*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件在访问之前检查文件或目录的状态，这会产生竞争条件，在执行访问之前可以用链接替换文件，从而导致软件访问错误的文件。

### Description:

The software checks the status of a file or directory before accessing it, which produces a race condition in which the file can be replaced with a link before the access is performed, causing the software to access the wrong file.

### 详细描述:

虽然开发人员可能期望在检查时间和使用时间之间存在非常窄的时间窗口，但仍存在竞争条件。攻击者可能导致软件速度变慢（例如，内存消耗），导致时间窗口变大。或者，在某些情况下，攻击者可以通过执行大量攻击来赢得比赛。

### Extended Description:

While developers might expect that there is a very narrow time window between the time of check and time of use, there is still a race condition. An attacker could cause the software to slow down (e.g. with memory consumption), causing the time window to become larger. Alternately, in some situations, the attacker could win the race by performing a large number of attacks.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这已经被“链接跟随”弱点（CWE-59）所涵盖。它包含在这里是因为很多人将竞争条件与链接问题联系起来;但是，并非所有链接以下问题都涉及竞争条件。

Weakness ID: 364

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Signal Handler Race Condition zh: -->信号处理器竞争条件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用引入竞争条件的信号处理程序。

### Description:

The software uses a signal handler that introduces a race condition.

### 详细描述:

竞争条件经常发生在信号处理程序中，因为信号处理程序支持异步动作。这些种族条件有各种根本原因和症状。攻击者可能能够利用信号处理程序竞争条件导致软件状态被破坏，可能导致拒绝服务甚至代码执行。  
当信号处理程序中发生非重入函数或状态敏感操作时，会发生这些问题，可以随时调用它们。这些行为可能违反被中断的“常规”代码或可能被调用的其他信号处理程序所做出的假设。如果在不合适的时刻调用这些函数 - 例如在非重入函数已经运行时 - 可能会发生内存损坏，这可能会被代码执行利用。通常发现的另一种信号竞争条件发生在信号处理程序中调用free时，导致双重释放，从而导致write-what-where条件。即使给定指针在释放后设置为NULL，在释放内存和指针设置为NULL之间仍然存在争用条件。如果为多个信号设置了相同的信号处理程序，则这尤其成问题 - 因为这意味着可以重新输入信号处理程序本身。  
有几个与信号处理程序相关的已知行为已经收到“信号处理程序竞争条件”的标签：  
  
  
信号处理程序和“常规”代码都可访问的共享状态（例如全局数据或静态变量）  
信号处理程序和其他信号处理程序之间的共享状态  
在信号处理程序中使用非重入功能 - 这通常意味着正在使用共享状态。例如，malloc（）和free（）是不可重入的，因为它们可能使用全局或静态数据结构来管理内存，并且它们被无辜的看似函数间接使用，例如syslog（）;这些函数可能被用于内存损坏，可能还有代码执行。  
将相同的信号处理函数与多个信号相关联 - 这可能意味着共享状态，因为访问了相同的代码和资源。例如，这可能是双重免费和使用后免费弱点的来源。  
使用setjmp和longjmp，或阻止信号处理程序将控制权返回到原始功能的其他机制  
虽然从技术上讲不是竞争条件，但是一些信号处理程序被设计为最多被调用一次，并且被调用不止一次会引入安全问题，即使没有任何并发​​调用信号处理程序。这可能是双重免费和使用后免费弱点的来源。  
  
  
信号处理程序漏洞通常基于缺少特定保护机制进行分类，尽管CWE不鼓励这种分类，因为程序员通常可以选择几种不同的机制来解决这些弱点。这种保护机制可以保留对共享资源的访问权限，以及相关代码的行为原子性：  
  
  
避免共享状态  
在信号处理程序中使用同步  
在常规代码中使用同步  
禁用或屏蔽其他信号，提供原子性（有效确保排他性）

### Extended Description:

Race conditions frequently occur in signal handlers, since signal handlers support asynchronous actions. These race conditions have a variety of root causes and symptoms. Attackers may be able to exploit a signal handler race condition to cause the software state to be corrupted, possibly leading to a denial of service or even code execution.  
These issues occur when non-reentrant functions, or state-sensitive actions occur in the signal handler, where they may be called at any time. These behaviors can violate assumptions being made by the "regular" code that is interrupted, or by other signal handlers that may also be invoked. If these functions are called at an inopportune moment - such as while a non-reentrant function is already running - memory corruption could occur that may be exploitable for code execution. Another signal race condition commonly found occurs when free is called within a signal handler, resulting in a double free and therefore a write-what-where condition. Even if a given pointer is set to NULL after it has been freed, a race condition still exists between the time the memory was freed and the pointer was set to NULL. This is especially problematic if the same signal handler has been set for more than one signal -- since it means that the signal handler itself may be reentered.  
There are several known behaviors related to signal handlers that have received the label of "signal handler race condition":  
  
  
Shared state (e.g. global data or static variables) that are accessible to both a signal handler and "regular" code  
Shared state between a signal handler and other signal handlers  
Use of non-reentrant functionality within a signal handler - which generally implies that shared state is being used. For example, malloc() and free() are non-reentrant because they may use global or static data structures for managing memory, and they are indirectly used by innocent-seeming functions such as syslog(); these functions could be exploited for memory corruption and, possibly, code execution.  
Association of the same signal handler function with multiple signals - which might imply shared state, since the same code and resources are accessed. For example, this can be a source of double-free and use-after-free weaknesses.  
Use of setjmp and longjmp, or other mechanisms that prevent a signal handler from returning control back to the original functionality  
While not technically a race condition, some signal handlers are designed to be called at most once, and being called more than once can introduce security problems, even when there are not any concurrent calls to the signal handler. This can be a source of double-free and use-after-free weaknesses.  
  
  
Signal handler vulnerabilities are often classified based on the absence of a specific protection mechanism, although this style of classification is discouraged in CWE because programmers often have a choice of several different mechanisms for addressing the weakness. Such protection mechanisms may preserve exclusivity of access to the shared resource, and behavioral atomicity for the relevant code:  
  
  
Avoiding shared state  
Using synchronization in the signal handler  
Using synchronization in the regular code  
Disabling or masking other signals, which provides atomicity (which effectively ensures exclusivity)

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可能未充分研究。

Weakness ID: 365

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Race Condition in Switch zh: -->交换机中的竞争条件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码包含一个switch语句，其中可以在交换机仍在执行时修改切换变量，从而导致意外行为。

### Description:

The code contains a switch statement in which the switched variable can be modified while the switch is still executing, resulting in unexpected behavior.

### 详细描述:

这个问题在涉及直通式案例陈述的switch语句中尤为重要 - 即那些不以break结尾的语句。如果交换机测试的变量在执行过程中发生变化，这可能会改变交换机的预期逻辑，以至于它将进程置于矛盾状态，在某些情况下甚至可能导致内存损坏。

### Extended Description:

This issue is particularly important in the case of switch statements that involve fall-through style case statements - ie., those which do not end with break. If the variable being tested by the switch changes in the course of execution, this could change the intended logic of the switch so much that it places the process in a contradictory state and in some cases could even result in memory corruption.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 366

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Race Condition within a Thread zh: -->线程中的竞争条件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果两个执行线程同时使用资源，则存在可能在无效时使用资源的可能性，从而使得执行状态未定义。

### Description:

If two threads of execution use a resource simultaneously, there exists the possibility that resources may be used while invalid, in turn making the state of execution undefined.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 367

提交日期 2008-10-14---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Time-of-check Time-of-use (TOCTOU) Race Condition zh: -->检查时间（TOCTOU）竞赛条件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件在使用该资源之前检查资源的状态，但资源的状态可以在检查和使用之间以一种使检查结果无效的方式发生变化。这可能导致软件在资源处于意外状态时执行无效操作。

### Description:

The software checks the state of a resource before using that resource, but the resource's state can change between the check and the use in a way that invalidates the results of the check. This can cause the software to perform invalid actions when the resource is in an unexpected state.

### 详细描述:

当攻击者可以在检查和使用之间影响资源状态时，这种弱点可能与安全相关。这可能发生在共享资源（如文件，内存，甚至是多线程程序中的变量）中。

### Extended Description:

This weakness can be security-relevant when an attacker can influence the state of the resource between check and use. This can happen with shared resources such as files, memory, or even variables in multithreaded programs.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

TOCTOU问题并不总是涉及符号链接，并不是每个符号链接问题都是TOCTOU问题。

Weakness ID: 368

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Context Switching Race Condition zh: -->上下文切换竞争条件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品执行一系列非原子操作以在跨越特权或其他安全边界的上下文之间切换，但竞争条件允许攻击者在切换期间修改或歪曲产品的行为。

### Description:

A product performs a series of non-atomic actions to switch between contexts that cross privilege or other security boundaries, but a race condition allows an attacker to modify or misrepresent the product's behavior during the switch.

### 详细描述:

这通常出现在Web浏览器漏洞中，攻击者可以在浏览器从受信任域转换到不受信任域时执行某些操作，反之亦然，浏览器使用信任级别和资源在一个域上执行操作。其他域名。

### Extended Description:

This is commonly seen in web browser vulnerabilities in which the attacker can perform certain actions while the browser is transitioning from a trusted to an untrusted domain, or vice versa, and the browser performs the actions on one domain using the trust level and resources of the other domain.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可以重叠信号处理器竞争条件。

Weakness ID: 369

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Divide By Zero zh: -->除以零*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品将值除以零。

### Description:

The product divides a value by zero.

### 详细描述:

当向产品提供意外值时，或者如果发生未正确检测到的错误，通常会发生此弱点。它经常发生在涉及物理尺寸的计算中，例如尺寸，长度，宽度和高度。

### Extended Description:

This weakness typically occurs when an unexpected value is provided to the product, or if an error occurs that is not properly detected. It frequently occurs in calculations involving physical dimensions such as size, length, width, and height.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 370

提交日期 2009-05-27---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Missing Check for Certificate Revocation after Initial Check zh: -->初始检查后缺少检查证书撤销*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件在初始撤销检查后不检查证书的撤销状态，这可能导致软件即使在稍后撤销证书后也执行特权操作。

### Description:

The software does not check the revocation status of a certificate after its initial revocation check, which can cause the software to perform privileged actions even after the certificate is revoked at a later time.

### 详细描述:

如果在每个需要权限的操作之前未检查证书的撤销状态，则系统可能会受到竞争条件的影响。如果在初始检查后撤销证书，则与撤销证书的所有者一起执行的所有后续操作都将失去证书保证的所有权益。实际上，几乎可以肯定使用撤销证书表示恶意活动。

### Extended Description:

If the revocation status of a certificate is not checked before each action that requires privileges, the system may be subject to a race condition. If a certificate is revoked after the initial check, all subsequent actions taken with the owner of the revoked certificate will lose all benefits guaranteed by the certificate. In fact, it is almost certain that the use of a revoked certificate indicates malicious activity.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 372

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Incomplete Internal State Distinction zh: -->不完整的内部状态区别*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件没有正确地确定它所处的状态，导致它假设它处于状态X，而实际上它处于状态Y，导致它以安全相关的方式执行不正确的操作。

### Description:

The software does not properly determine which state it is in, causing it to assume it is in state X when in fact it is in state Y, causing it to perform incorrect operations in a security-relevant manner.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这在概念上与其他类别重叠，例如验证不充分，但此条目是指产品对其自身状态的错误认知。

Weakness ID: 373

提交日期 2010-12-13---> 修改日期 2010-12-13

* **Weakness Name:** *en: --> DEPRECATED: State Synchronization Error zh: -->DEPRECATED：状态同步错误*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此条目已弃用，因为它与竞争条件（CWE-362）和不正确同步（CWE-662）相同的概念重叠。

### Description:

This entry was deprecated because it overlapped the same concepts as race condition (CWE-362) and Improper Synchronization (CWE-662).

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 374

提交日期 2010-06-21---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Passing Mutable Objects to an Untrusted Method zh: -->将可变对象传递给不受信任的方法*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

程序将非克隆的可变数据作为参数发送给方法或函数。

### Description:

The program sends non-cloned mutable data as an argument to a method or function.

### 详细描述:

已调用的函数或方法可以更改或删除可变数据。这可能违反了调用函数对其状态所做的假设。在通过引用可变数据调用未知代码的情况下，此外部代码可以对发送的数据进行更改。如果先前未克隆此数据，则修改后的数据在执行上下文中可能无效。

### Extended Description:

The function or method that has been called can alter or delete the mutable data. This could violate assumptions that the calling function has made about its state. In situations where unknown code is called with references to mutable data, this external code could make changes to the data sent. If this data was not previously cloned, the modified data might not be valid in the context of execution.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 375

提交日期 2010-09-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Returning a Mutable Object to an Untrusted Caller zh: -->将可变对象返回给不受信任的调用者*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

将非克隆的可变数据作为返回值发送可能导致该数据被调用函数更改或删除。

### Description:

Sending non-cloned mutable data as a return value may result in that data being altered or deleted by the calling function.

### 详细描述:

在函数返回对可变数据的引用的情况下，调用该函数的外部代码可能会对发送的数据进行更改。如果此数据先前未被克隆，则该类将使用可能违反其内部状态假设的修改数据。

### Extended Description:

In situations where functions return references to mutable data, it is possible that the external code which called the function may make changes to the data sent. If this data was not previously cloned, the class will then be using modified data which may violate assumptions about its internal state.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 378

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Creation of Temporary File With Insecure Permissions zh: -->使用不安全的权限创建临时文件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在没有适当措施或控制的情况下打开临时文件可能会使文件，其内容以及任何易受攻击的功能受到攻击。

### Description:

Opening temporary files without appropriate measures or controls can leave the file, its contents and any function that it impacts vulnerable to attack.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 379

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Creation of Temporary File in Directory with Incorrect Permissions zh: -->在具有不正确权限的目录中创建临时文件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在目录中创建一个临时文件，其权限允许非预期的actor确定文件是否存在或以其他方式访问该文件。

### Description:

The software creates a temporary file in a directory whose permissions allow unintended actors to determine the file's existence or otherwise access that file.

### 详细描述:

在某些操作系统上，对于具有足够权限访问该目录的任何用户，临时文件存在的事实可能是显而易见的。由于文件可见，因此可以知道使用临时文件的应用程序。如果有权访问系统上的进程列表，则攻击者已获得有关用户当时正在执行的操作的信息。通过将此与用户运行的应用程序相关联，攻击者可能会发现用户的操作是什么。由此可以破坏更高级别的安全性。

### Extended Description:

On some operating systems, the fact that the temporary file exists may be apparent to any user with sufficient privileges to access that directory. Since the file is visible, the application that is using the temporary file could be known. If one has access to list the processes on the system, the attacker has gained information about what the user is doing at that time. By correlating this with the applications the user is running, an attacker could potentially discover what a user's actions are. From this, higher levels of security could be breached.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 382

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> J2EE Bad Practices: Use of System.exit() zh: -->J2EE不良做法：使用System.exit（）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

J2EE应用程序使用System.exit（），它也会关闭其容器。

### Description:

A J2EE application uses System.exit(), which also shuts down its container.

### 详细描述:

Web应用程序尝试关闭应用程序容器绝不是一个好主意。访问可以关闭应用程序的功能是拒绝服务（DoS）攻击的途径。

### Extended Description:

It is never a good idea for a web application to attempt to shut down the application container. Access to a function that can shut down the application is an avenue for Denial of Service (DoS) attacks.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 383

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> J2EE Bad Practices: Direct Use of Threads zh: -->J2EE不良做法：直接使用线程*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在某些情况下，Web应用程序中的线程管理是被禁止的，并且总是非常容易出错。

### Description:

Thread management in a Web application is forbidden in some circumstances and is always highly error prone.

### 详细描述:

在某些情况下，J2EE标准禁止Web应用程序中的线程管理，并且总是非常容易出错。管理线程很困难，并且可能以不可预测的方式干扰应用程序容器的行为。即使不干扰容器，线程管理通常也会导致难以检测和诊断的错误，如死锁，竞争条件和其他同步错误。

### Extended Description:

Thread management in a web application is forbidden by the J2EE standard in some circumstances and is always highly error prone. Managing threads is difficult and is likely to interfere in unpredictable ways with the behavior of the application container. Even without interfering with the container, thread management usually leads to bugs that are hard to detect and diagnose like deadlock, race conditions, and other synchronization errors.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 384

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Session Fixation zh: -->会话固定*
* **Abstraction:** *Compound* **Structure:** *Composite* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在不使任何现有会话标识符无效的情况下认证用户或以其他方式建立新用户会话使攻击者有机会窃取经认证的会话。

### Description:

Authenticating a user, or otherwise establishing a new user session, without invalidating any existing session identifier gives an attacker the opportunity to steal authenticated sessions.

### 详细描述:

在以下情况下通常会出现这种情况：  
  
1. Web应用程序在不首先使现有会话无效的情况下对用户进行认证，从而继续使用已经与该用户相关联的会话。  
2.攻击者能够强制用户使用已知的会话标识符，以便一旦用户进行身份验证，攻击者就可以访问经过身份验证的会话。  
3.应用程序或容器使用可预测的会话标识符。在会话固定漏洞的一般利用中，攻击者在Web应用程序上创建新会话并记录关联的会话标识符。然后，攻击者使用该会话标识符使受害者与服务器关联并可能进行身份验证，从而使攻击者通过活动会话访问用户的帐户。

### Extended Description:

Such a scenario is commonly observed when:  
  
1. A web application authenticates a user without first invalidating the existing session, thereby continuing to use the session already associated with the user.  
2. An attacker is able to force a known session identifier on a user so that, once the user authenticates, the attacker has access to the authenticated session.  
3. The application or container uses predictable session identifiers. In the generic exploit of session fixation vulnerabilities, an attacker creates a new session on a web application and records the associated session identifier. The attacker then causes the victim to associate, and possibly authenticate, against the server using that session identifier, giving the attacker access to the user's account through the active session.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

其他攻击媒介包括DNS中毒和相关的基于网络的攻击，其中攻击者通过重定向对有效站点的请求来使用户访问恶意站点。基于网络的攻击通常涉及受害者网络上的物理存在或网络上受感染机器的控制，这使得它们更难以远程利用，但不应忽视它们的重要性。不太安全的会话管理机制，例如Apache Tomcat中的默认实现，允许在URL上指定通常在cookie中预期的会话标识符，这使得攻击者能够通过电子邮件发送给受害者使用固定的会话标识符。恶意URL。

Weakness ID: 385

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Covert Timing Channel zh: -->隐蔽时间通道*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

隐蔽定时信道通过随时间调整系统行为的某些方面来传达信息，使得接收信息的程序可以观察系统行为并推断受保护的信息。

### Description:

Covert timing channels convey information by modulating some aspect of system behavior over time, so that the program receiving the information can observe system behavior and infer protected information.

### 详细描述:

在某些情况下，知道何时在各方之间传输数据可以向恶意用户提供特权信息。此外，从外部监视操作的时间可能会泄露敏感数据。例如，如果执行操作所花费的时间根据状态而变化，则加密操作可以暴露其内部状态。  
隐蔽信道通常被分类为存储或定时信道。隐蔽定时信道的一些示例是系统的寻呼速率，特定事务执行所需的时间以及获得对共享总线的访问所花费的时间。

### Extended Description:

In some instances, knowing when data is transmitted between parties can provide a malicious user with privileged information. Also, externally monitoring the timing of operations can potentially reveal sensitive data. For example, a cryptographic operation can expose its internal state if the time it takes to perform the operation varies, based on the state.  
Covert channels are frequently classified as either storage or timing channels. Some examples of covert timing channels are the system's paging rate, the time a certain transaction requires to execute, and the time it takes to gain access to a shared bus.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 386

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Symbolic Name not Mapping to Correct Object zh: -->符号名称不映射到正确的对象*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

使用对象的常量符号引用，即使引用可以随时间解析为不同的对象。

### Description:

A constant symbolic reference to an object is used, even though the reference can resolve to a different object over time.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 390

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Detection of Error Condition Without Action zh: -->无动作的错误条件检测*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件检测到特定错误，但不采取任何操作来处理错误。

### Description:

The software detects a specific error, but takes no actions to handle the error.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 391

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Unchecked Error Condition zh: -->未选中的错误条件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

忽略异常和其他错误条件可能允许攻击者引起意外行为而不被注意。

### Description:

Ignoring exceptions and other error conditions may allow an attacker to induce unexpected behavior unnoticed.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目需要进行重大修改。它目前结合了来自三种不同分类法的信息，但每种分类法都在讨论一个略有不同的问题。

Weakness ID: 392

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Missing Report of Error Condition zh: -->错误报告错误情况*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件遇到错误但未提供状态代码或返回值以指示发生了错误。

### Description:

The software encounters an error but does not provide a status code or return value to indicate that an error has occurred.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 393

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Return of Wrong Status Code zh: -->返回错误的状态代码*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

函数或操作返回不正确的返回值或状态代码，但不会指示错误，但会导致产品根据错误的结果修改其行为。

### Description:

A function or operation returns an incorrect return value or status code that does not indicate an error, but causes the product to modify its behavior based on the incorrect result.

### 详细描述:

这可能导致不可预测的行为。如果该函数用于制定安全关键决策或提供安全关键信息，则错误的状态代码可能导致软件认为某个操作是安全的，即使它不是。

### Extended Description:

This can lead to unpredictable behavior. If the function is used to make security-critical decisions or provide security-critical information, then the wrong status code can cause the software to assume that an action is safe, even when it is not.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能是主要的或结果，但它可能是其他问题的主要原因。

Weakness ID: 394

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Unexpected Status Code or Return Value zh: -->意外的状态代码或返回值*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件无法正确检查函数或操作何时返回对函数合法的值，但软件不期望该值。

### Description:

The software does not properly check when a function or operation returns a value that is legitimate for the function, but is not expected by the software.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

通常是主要的，但可能是行为改变或API滥用等问题的结果。这可能会产生漏洞。

Weakness ID: 395

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of NullPointerException Catch to Detect NULL Pointer Dereference zh: -->使用NullPointerException Catch来检测空指针解除引用*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

捕获NullPointerException不应用作编程检查的替代方法，以防止取消引用空指针。

### Description:

Catching NullPointerException should not be used as an alternative to programmatic checks to prevent dereferencing a null pointer.

### 详细描述:

程序员通常在三种情况下捕获NullPointerException：  
  
  
该程序包含空指针取消引用。捕获生成的异常比修复底层问题更容易。  
程序显式抛出NullPointerException来发出错误信号。  
该代码是测试工具的一部分，为测试中的类提供意外输入。  
  
  
在这三种情况中，只有最后一种是可以接受的。

### Extended Description:

Programmers typically catch NullPointerException under three circumstances:  
  
  
The program contains a null pointer dereference. Catching the resulting exception was easier than fixing the underlying problem.  
The program explicitly throws a NullPointerException to signal an error condition.  
The code is part of a test harness that supplies unexpected input to the classes under test.  
  
  
Of these three circumstances, only the last is acceptable.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 396

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Declaration of Catch for Generic Exception zh: -->通用例外的捕获声明*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

捕获过于广泛的异常会促使复杂的错误处理代码更容易包含安全漏洞。

### Description:

Catching overly broad exceptions promotes complex error handling code that is more likely to contain security vulnerabilities.

### 详细描述:

多个catch块可能会变得丑陋和重复，但是通过捕获像Exception这样的高级类来“压缩”catch块会掩盖需要特殊处理的异常或者在程序中此时不应该捕获的异常。捕获过于宽泛的异常基本上会破坏Java类型异常的目的，如果程序增长并开始抛出新类型的异常，则会变得特别危险。新的异常类型不会受到任何关注。

### Extended Description:

Multiple catch blocks can get ugly and repetitive, but "condensing" catch blocks by catching a high-level class like Exception can obscure exceptions that deserve special treatment or that should not be caught at this point in the program. Catching an overly broad exception essentially defeats the purpose of Java's typed exceptions, and can become particularly dangerous if the program grows and begins to throw new types of exceptions. The new exception types will not receive any attention.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 397

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Declaration of Throws for Generic Exception zh: -->通用例外的抛出声明*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

抛出过于广泛的异常会促使复杂的错误处理代码更容易包含安全漏洞。

### Description:

Throwing overly broad exceptions promotes complex error handling code that is more likely to contain security vulnerabilities.

### 详细描述:

声明抛出Exception或Throwable的方法会使调用者难以执行正确的错误处理和错误恢复。例如，Java的异常机制被设置为使调用者能够轻松地预测可能出错的内容并编写代码来处理每个特定的异常情况。声明方法抛出一般形式的异常会使该系统失效。

### Extended Description:

Declaring a method to throw Exception or Throwable makes it difficult for callers to perform proper error handling and error recovery. Java's exception mechanism, for example, is set up to make it easy for callers to anticipate what can go wrong and write code to handle each specific exceptional circumstance. Declaring that a method throws a generic form of exception defeats this system.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

对于C ++，这个弱点仅适用于C ++ 98，C ++ 03和C ++ 11。它依赖于一种称为动态异常规范的功能，它是早期版本的C ++的一部分，但在C ++ 11中已被弃用。它已被删除C ++ 17及更高版本。

Weakness ID: 400

提交日期 2008-10-14---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Uncontrolled Resource Consumption zh: -->不受控制的资源消耗*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件没有适当地控制有限资源的分配和维护，从而使行动者能够影响所消耗的资源量，最终导致可用资源的耗尽。

### Description:

The software does not properly control the allocation and maintence of a limited resource thereby enabling an actor to influence the amount of resources consumed, eventually leading to the exhaustion of available resources.

### 详细描述:

有限的资源包括内存，文件系统存储，数据库连接池条目和CPU。如果攻击者可以触发这些有限资源的分配，但资源的数量或大小未受控制，则攻击者可能会导致拒绝服务，从而消耗所有可用资源。这将阻止有效用户访问该软件，并且可能对周围环境产生影响。例如，针对应用程序的内存耗尽攻击可能会降低应用程序及其主机操作系统的速度。  
至少有三种不同的情况通常会导致资源耗尽：  
  
  
对分配的资源数量缺乏限制  
在到达关闭阶段之前丢失对资源的所有引用  
处理后没有关闭/返回资源  
  
  
资源耗尽问题通常是由于以下情况的错误实施导致的：  
  
  
错误条件和其他特殊情况。  
混淆程序的哪个部分负责释放资源。

### Extended Description:

Limited resources include memory, file system storage, database connection pool entries, and CPU. If an attacker can trigger the allocation of these limited resources, but the number or size of the resources is not controlled, then the attacker could cause a denial of service that consumes all available resources. This would prevent valid users from accessing the software, and it could potentially have an impact on the surrounding environment. For example, a memory exhaustion attack against an application could slow down the application as well as its host operating system.  
There are at least three distinct scenarios which can commonly lead to resource exhaustion:  
  
  
Lack of throttling for the number of allocated recources  
Losing all references to a resource before reaching the shutdown stage  
Not closing/returning resource after processing  
  
  
Resource exhaustion problems are often result due to an incorrect implementation of the following situations:  
  
  
Error conditions and other exceptional circumstances.  
Confusion over which part of the program is responsible for releasing the resource.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

漏洞理论主要是关于行为和资源如何相互作用。根据观点，“资源耗尽”可视为后果或攻击。此条目试图反映导致这些攻击（或后果）发生的潜在弱点。

Weakness ID: 401

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Release of Memory Before Removing Last Reference zh: -->删除最后一个引用之前内存释放不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件在使用后无法充分跟踪和释放已分配的内存，这会慢慢消耗剩余的内存。

### Description:

The software does not sufficiently track and release allocated memory after it has been used, which slowly consumes remaining memory.

### 详细描述:

这通常是由于错误处理格式错误的数据或意外中断的会话而触发的。

### Extended Description:

This is often triggered by improper handling of malformed data or unexpectedly interrupted sessions.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这通常是由于不正确处理格式错误的数据或提前终止会话而导致的弱点。

Weakness ID: 402

提交日期 2008-04-11---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Transmission of Private Resources into a New Sphere ('Resource Leak') zh: -->私人资源进入新领域（'资源泄漏'）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当这些资源仅供软件访问时，该软件可为不受信任的各方提供资源。

### Description:

The software makes resources available to untrusted parties when those resources are only intended to be accessed by the software.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 403

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Exposure of File Descriptor to Unintended Control Sphere ('File Descriptor Leak') zh: -->文件描述符暴露于非预期控制球（'文件描述符泄漏'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在调用子进程之前，进程不会关闭敏感文件描述符，这允许子进程使用这些描述符执行未经授权的I / O操作。

### Description:

A process does not close sensitive file descriptors before invoking a child process, which allows the child to perform unauthorized I/O operations using those descriptors.

### 详细描述:

分叉或执行新进程时，子进程继承任何打开的文件描述符。当子进程的权限少于父进程时，如果子进程可以访问文件描述符但没有访问关联文件的权限，则可能会引入漏洞。

### Extended Description:

When a new process is forked or executed, the child process inherits any open file descriptors. When the child process has fewer privileges than the parent process, this might introduce a vulnerability if the child process can access the file descriptor but does not have the privileges to access the associated file.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 404

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Resource Shutdown or Release zh: -->资源关闭或发布不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

程序在可用于重用之前不会释放或错误地释放资源。

### Description:

The program does not release or incorrectly releases a resource before it is made available for re-use.

### 详细描述:

创建或分配资源时，开发人员负责正确释放资源以及计算所有可能的到期或失效路径，例如设定的时间段或撤销。

### Extended Description:

When a resource is created or allocated, the developer is responsible for properly releasing the resource as well as accounting for all potential paths of expiration or invalidation, such as a set period of time or revocation.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠内存泄漏，不对称的资源消耗，错误的输入错误。

Weakness ID: 405

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Asymmetric Resource Consumption (Amplification) zh: -->不对称资源消耗（扩增）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

未适当监视或控制资源消耗的软件可能导致系统性能下降。

### Description:

Software that does not appropriately monitor or control resource consumption can lead to adverse system performance.

### 详细描述:

如果软件允许恶意用户或攻击者消耗比访问级别允许的更多资源，则这种情况会被放大。利用这种弱点可能导致不对称的资源消耗，有助于对系统或网络的放大攻击。

### Extended Description:

This situation is amplified if the software allows malicious users or attackers to consume more resources than their access level permits. Exploiting such a weakness can lead to asymmetric resource consumption, aiding in amplification attacks against the system or the network.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 406

提交日期 2008-10-14---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Insufficient Control of Network Message Volume (Network Amplification) zh: -->网络消息量控制不足（网络扩容）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不能充分监视或控制传输的网络流量，因此参与者可以使软件传输的流量超过该参与者应允许的流量。

### Description:

The software does not sufficiently monitor or control transmitted network traffic volume, so that an actor can cause the software to transmit more traffic than should be allowed for that actor.

### 详细描述:

在缺乏限制非对称资源消耗的策略的情况下，应用程序或系统无法区分合法传输和旨在用作对目标系统的放大攻击的流量。系统通常可以配置为根据客户端的来源或访问级别限制代表客户端发送的流量。这通常在资源分配策略中定义。在没有跟踪传输的机制的情况下，系统或应用程序可以容易地被滥用以传输比应该允许的请求或客户端更不对称更大的流量。

### Extended Description:

In the absence of a policy to restrict asymmetric resource consumption, the application or system cannot distinguish between legitimate transmissions and traffic intended to serve as an amplifying attack on target systems. Systems can often be configured to restrict the amount of traffic sent out on behalf of a client, based on the client's origin or access level. This is usually defined in a resource allocation policy. In the absence of a mechanism to keep track of transmissions, the system or application can be easily abused to transmit asymmetrically greater traffic than the request or client should be permitted to.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能是由于简化欺骗攻击的弱点所致。

Weakness ID: 407

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Algorithmic Complexity zh: -->算法复杂性*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品中的算法具有低效的最坏情况计算复杂度，可能对系统性能有害并且可以由攻击者触发，通常使用精心设计的操作来确保达到最坏的情况。

### Description:

An algorithm in a product has an inefficient worst-case computational complexity that may be detrimental to system performance and can be triggered by an attacker, typically using crafted manipulations that ensure that the worst case is being reached.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 408

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incorrect Behavior Order: Early Amplification zh: -->行为顺序不正确：早期扩增*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件允许实体在进行认证或授权之前执行合法但昂贵的操作。

### Description:

The software allows an entity to perform a legitimate but expensive operation before authentication or authorization has taken place.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠身份验证错误。

Weakness ID: 409

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Handling of Highly Compressed Data (Data Amplification) zh: -->高压缩数据处理不当（数据放大）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法处理或错误处理具有非常高压缩比的压缩输入，从而产生大量输出。

### Description:

The software does not handle or incorrectly handles a compressed input with a very high compression ratio that produces a large output.

### 详细描述:

数据放大的一个例子是“解压缩炸弹”，一种小型ZIP文件，在解压缩时可以产生大量数据。

### Extended Description:

An example of data amplification is a "decompression bomb," a small ZIP file that can produce a large amount of data when it is decompressed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 410

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Insufficient Resource Pool zh: -->资源池不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件的资源池不足以处理峰值需求，这允许攻击者通过使用（相对）大量的资源请求来阻止其他人访问资源。

### Description:

The software's resource pool is not large enough to handle peak demand, which allows an attacker to prevent others from accessing the resource by using a (relatively) large number of requests for resources.

### 详细描述:

通常结果是连接或会话的“泛滥”。

### Extended Description:

Frequently the consequence is a "flood" of connection or sessions.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 412

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Unrestricted Externally Accessible Lock zh: -->无限制的外部可访问锁*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件正确地检查是否存在锁，但是锁可以由外部控制或受到在预期控制范围之外的行为者的影响。

### Description:

The software properly checks for the existence of a lock, but the lock can be externally controlled or influenced by an actor that is outside of the intended sphere of control.

### 详细描述:

这可以防止软件对相关资源起作用或执行由锁的存在控制的其他行为。相关锁可能包括独占锁或互斥锁，或修改被视为锁的共享资源。如果锁定可以无限期保留，那么拒绝服务可能是永久性的。

### Extended Description:

This prevents the software from acting on associated resources or performing other behaviors that are controlled by the presence of the lock. Relevant locks might include an exclusive lock or mutex, or modifying a shared resource that is treated as a lock. If the lock can be held for an indefinite period of time, then the denial of service could be permanent.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

当“池”大小为1时，这与资源池不足重叠。它也可能是竞争条件的结果，尽管在某些情况下时间窗口可能非常大。

Weakness ID: 413

提交日期 2010-09-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Resource Locking zh: -->资源锁定不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当软件必须具有对资源的独占访问权限时，软件不会锁定或未正确锁定资源。

### Description:

The software does not lock or does not correctly lock a resource when the software must have exclusive access to the resource.

### 详细描述:

当资源未正确锁定时，攻击者可以在软件操作资源时修改资源。这可能违反了软件假设资源不会改变，可能导致意外行为的假设。

### Extended Description:

When a resource is not properly locked, an attacker could modify the resource while it is being operated on by the software. This might violate the software's assumption that the resource will not change, potentially leading to unexpected behaviors.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 414

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Missing Lock Check zh: -->缺少锁定检查*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在对资源执行敏感操作之前，产品不会检查是否存在锁定。

### Description:

A product does not check to see if a lock is present before performing sensitive operations on a resource.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 415

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Double Free zh: -->双免费*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品在同一内存地址上调用free（）两次，可能导致意外内存位置的修改。

### Description:

The product calls free() twice on the same memory address, potentially leading to modification of unexpected memory locations.

### 详细描述:

当程序使用相同的参数调用free（）两次时，程序的内存管理数据结构将被破坏。这种损坏可能导致程序崩溃，或者在某些情况下导致两个以后的malloc（）调用返回相同的指针。如果malloc（）返回相同的值两次，并且程序稍后让攻击者控制写入这个双重分配的内存的数据，则该程序容易受到缓冲区溢出攻击。

### Extended Description:

When a program calls free() twice with the same argument, the program's memory management data structures become corrupted. This corruption can cause the program to crash or, in some circumstances, cause two later calls to malloc() to return the same pointer. If malloc() returns the same value twice and the program later gives the attacker control over the data that is written into this doubly-allocated memory, the program becomes vulnerable to a buffer overflow attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这通常是由另一个弱点引起的，例如线程之间的未处理错误或竞争条件。它也可能是缓冲区溢出等弱点的主要原因。

Weakness ID: 416

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use After Free zh: -->免费使用*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在释放内存后引用内存可能导致程序崩溃，使用意外值或执行代码。

### Description:

Referencing memory after it has been freed can cause a program to crash, use unexpected values, or execute code.

### 详细描述:

使用先前释放的内存可能会产生任何数量的不利后果，从有效数据的损坏到任意代码的执行，具体取决于缺陷的实例化和时间安排。可能发生数据损坏的最简单方法是系统重用已释放的内存。释放后使用错误有两个常见且有时重叠的原因：  
  
  
错误条件和其他特殊情况。  
混淆程序的哪个部分负责释放内存。  
  
  
在这种情况下，有问题的内存在释放后的某个时刻有效地分配给另一个指针。再次使用指向释放内存的原始指针，并指向新分配中的某个位置。随着数据的改变，它会破坏有效使用的内存;这会导致过程中出现未定义的行为。  
如果新分配的数据有机会保存类，例如在C ++中，各种函数指针可能分散在堆数据中。如果用有效shellcode的地址覆盖这些函数指针之一，则可以实现任意代码的执行。

### Extended Description:

The use of previously-freed memory can have any number of adverse consequences, ranging from the corruption of valid data to the execution of arbitrary code, depending on the instantiation and timing of the flaw. The simplest way data corruption may occur involves the system's reuse of the freed memory. Use-after-free errors have two common and sometimes overlapping causes:  
  
  
Error conditions and other exceptional circumstances.  
Confusion over which part of the program is responsible for freeing the memory.  
  
  
In this scenario, the memory in question is allocated to another pointer validly at some point after it has been freed. The original pointer to the freed memory is used again and points to somewhere within the new allocation. As the data is changed, it corrupts the validly used memory; this induces undefined behavior in the process.  
If the newly allocated data chances to hold a class, in C++ for example, various function pointers may be scattered within the heap data. If one of these function pointers is overwritten with an address to valid shellcode, execution of arbitrary code can be achieved.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 419

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Unprotected Primary Channel zh: -->不受保护的主要频道*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用主要通道进行管理或限制功能，但它不能正确保护通道。

### Description:

The software uses a primary channel for administration or restricted functionality, but it does not properly protect the channel.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 420

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Unprotected Alternate Channel zh: -->不受保护的替代频道*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件保护主要通道，但它不对备用通道使用相同级别的保护。

### Description:

The software protects a primary channel, but it does not use the same level of protection for an alternate channel.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能是身份验证错误的主要原因，也是未处理错误情况的结果。

Weakness ID: 421

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Race Condition During Access to Alternate Channel zh: -->访问备用频道期间的竞争条件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品打开一个备用频道与授权用户进行通信，但其他演员可以访问该频道。

### Description:

The product opens an alternate channel to communicate with an authorized user, but the channel is accessible to other actors.

### 详细描述:

这会创建一个竞争条件，允许攻击者在授权用户之前访问该通道。

### Extended Description:

This creates a race condition that allows an attacker to access the channel before the authorized user does.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 422

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Unprotected Windows Messaging Channel ('Shatter') zh: -->不受保护的Windows消息传递通道（'Shatter'）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在以提升的权限运行时，软件无法正确验证Windows Messaging System中的消息来源，从而创建了一个备用通道，攻击者可以通过该通道直接向产品发送消息。

### Description:

The software does not properly verify the source of a message in the Windows Messaging System while running at elevated privileges, creating an alternate channel through which an attacker can directly send a message to the product.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠权限错误和UI错误。

Weakness ID: 423

提交日期 2008-11-24---> 修改日期 2009-10-29

* **Weakness Name:** *en: --> DEPRECATED (Duplicate): Proxied Trusted Channel zh: -->弃用（重复）：代理可信通道*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此条目已被弃用，因为它与CWE-441重复。所有内容均已转移至CWE-441。

### Description:

This entry has been deprecated because it was a duplicate of CWE-441. All content has been transferred to CWE-441.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 424

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Protection of Alternate Path zh: -->替代路径的保护不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品不足以保护用户可以用来访问受限功能或资源的所有可能路径。

### Description:

The product does not sufficiently protect all possible paths that a user can take to access restricted functionality or resources.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 425

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Direct Request ('Forced Browsing') zh: -->直接请求（'强制浏览'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序未对所有受限制的URL，脚本或文件充分执行适当的授权。

### Description:

The web application does not adequately enforce appropriate authorization on all restricted URLs, scripts, or files.

### 详细描述:

易受直接请求攻击影响的Web应用程序通常会假设这样的资源只能通过给定的导航路径到达，因此只能在路径中的某些点应用授权。

### Extended Description:

Web applications susceptible to direct request attacks often make the false assumption that such resources can only be reached through a given navigation path and so only apply authorization at certain points in the path.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠修改假定不可变数据（MAID），授权错误，容器错误;通常是主要的其他弱点，如XSS和SQL注入。

Weakness ID: 426

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Untrusted Search Path zh: -->不受信任的搜索路径*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序使用外部提供的搜索路径搜索关键资源，该搜索路径可以指向不在应用程序直接控制之下的资源。

### Description:

The application searches for critical resources using an externally-supplied search path that can point to resources that are not under the application's direct control.

### 详细描述:

这可能允许攻击者执行自己的程序，访问未经授权的数据文件或以意外方式修改配置。如果应用程序使用搜索路径来定位关键资源（如程序），则攻击者可以修改该搜索路径以指向恶意程序，然后目标应用程序将执行该程序。问题扩展到应用程序信任的任何类型的关键资源。  
一些最常见的不受信任的搜索路径变体是：  
  
  
在各种基于UNIX和Linux的系统中，可以查询PATH环境变量以定位可执行程序，并且可以使用LD\_PRELOAD来定位单独的库。  
在各种基于Microsoft的系统中，如果在搜索顺序中较早出现的其他路径中找不到DLL，则查询PATH环境变量以定位DLL。

### Extended Description:

This might allow attackers to execute their own programs, access unauthorized data files, or modify configuration in unexpected ways. If the application uses a search path to locate critical resources such as programs, then an attacker could modify that search path to point to a malicious program, which the targeted application would then execute. The problem extends to any type of critical resource that the application trusts.  
Some of the most common variants of untrusted search path are:  
  
  
In various UNIX and Linux-based systems, the PATH environment variable may be consulted to locate executable programs, and LD\_PRELOAD may be used to locate a separate library.  
In various Microsoft-based systems, the PATH environment variable is consulted to locate a DLL, if the DLL is not found in other paths that appear earlier in the search order.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

Windows上的搜索路径问题未得到充分研究，可能报告不足。

Weakness ID: 427

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Uncontrolled Search Path Element zh: -->不受控制的搜索路径元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用固定或受控的搜索路径来查找资源，但该路径中的一个或多个位置可能受非预期参与者的控制。

### Description:

The product uses a fixed or controlled search path to find resources, but one or more locations in that path can be under the control of unintended actors.

### 详细描述:

虽然这种弱点可能出现在任何类型的资源中，但是当产品使用目录搜索路径查找可执行文件或代码库时，它经常被引入，但路径包含可由攻击者修改的目录，例如“/ tmp”或当前的工作目录。  
在基于Windows的系统中，当使用不包含完全限定路径的DLL名称调用LoadLibrary或LoadLibraryEx函数时，该函数遵循包含两个可能不受控制的路径元素的搜索顺序：  
  
  
加载程序的目录  
当前的工作目录。  
  
  
在某些情况下，攻击可以远程进行，例如使用SMB或WebDAV网络共享时。  
在一些基于Unix的系统中，可能会创建一个包含空元素的PATH，例如通过将空变量拼接到PATH中。此空元素可以解释为等效于当前工作目录，该目录可能是不受信任的搜索元素。

### Extended Description:

Although this weakness can occur with any type of resource, it is frequently introduced when a product uses a directory search path to find executables or code libraries, but the path contains a directory that can be modified by an attacker, such as "/tmp" or the current working directory.  
In Windows-based systems, when the LoadLibrary or LoadLibraryEx function is called with a DLL name that does not contain a fully qualified path, the function follows a search order that includes two path elements that might be uncontrolled:  
  
  
the directory from which the program has been loaded  
the current working directory.  
  
  
In some cases, the attack can be conducted remotely, such as when SMB or WebDAV network shares are used.  
In some Unix-based systems, a PATH might be created that contains an empty element, e.g. by splicing an empty variable into the PATH. This empty element can be interpreted as equivalent to the current working directory, which might be an untrusted search element.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

与不受信任的搜索路径（CWE-426）不同，该路径固有地涉及对控制球体的定义的控制（即，修改搜索路径），该条目涉及固定的控制球体，其中球体的某些部分可能受到攻击者控制（即，攻击者无法修改搜索路径，但路径的一个元素可能受攻击者控制）。

Weakness ID: 428

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Unquoted Search Path or Element zh: -->不带引号的搜索路径或元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用包含不带引号的元素的搜索路径，其中元素包含空格或其他分隔符。这可能导致产品访问父路径中的资源。

### Description:

The product uses a search path that contains an unquoted element, in which the element contains whitespace or other separators. This can cause the product to access resources in a parent path.

### 详细描述:

如果恶意个人有权访问文件系统，则可以通过插入“C：\ Program.exe”这样的文件来提升权限，以便由使用WinExec的特权程序运行。

### Extended Description:

If a malicious individual has access to the file system, it is possible to elevate privileges by inserting such a file as "C:\Program.exe" to be run by a privileged program making use of WinExec.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这种弱点可能适用于任何支持文件名中的空格的操作系统，尤其是任何使用户可以轻松地将空格插入文件名或文件夹（例如Windows）的操作系统。虽然Unix在技术上支持空间，但通常可以避免这种做法。 。

Weakness ID: 430

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Deployment of Wrong Handler zh: -->部署错误的处理程序*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

分配了错误的“处理程序”来处理对象。

### Description:

The wrong "handler" is assigned to process an object.

### 详细描述:

部署错误处理程序的一个示例是调用servlet来显示.JSP文件的源代码，或者自动“确定”对象的类型，即使它与明确指定的类型相矛盾。

### Extended Description:

An example of deploying the wrong handler would be calling a servlet to reveal source code of a .JSP file, or automatically "determining" type of the object even if it is contradictory to an explicitly specified type.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 431

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Missing Handler zh: -->缺少处理程序*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

处理程序不可用或未实现。

### Description:

A handler is not available or implemented.

### 详细描述:

当抛出异常并且未捕获异常时，该过程放弃了决定给定故障或事件是否值得执行更改的机会。

### Extended Description:

When an exception is thrown and not caught, the process has given up an opportunity to decide if a given failure or event is worth a change in execution.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 432

提交日期 2008-04-11---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Dangerous Signal Handler not Disabled During Sensitive Operations zh: -->敏感操作期间未禁用危险信号处理程序*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序使用与其他信号处理程序共享状态的信号处理程序，但在原始信号处理程序仍在运行时，它无法正确屏蔽或阻止调用这些信号处理程序。

### Description:

The application uses a signal handler that shares state with other signal handlers, but it does not properly mask or prevent those signal handlers from being invoked while the original signal handler is still running.

### 详细描述:

在执行信号处理程序期间，当发送不同的信号时，它可以被另一个处理程序中断。如果两个处理程序共享状态 - 例如全局变量 - 那么攻击者可以通过在第一个处理程序完成执行之前发送另一个信号来破坏状态。

### Extended Description:

During the execution of a signal handler, it can be interrupted by another handler when a different signal is sent. If the two handlers share state - such as global variables - then an attacker can corrupt the state by sending another signal before the first handler has completed execution.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 433

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Unparsed Raw Web Content Delivery zh: -->Unparsed Raw Web Content Delivery*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件将原始内容或支持代码存储在Web文档根目录下，其扩展名不是由服务器专门处理的。

### Description:

The software stores raw content or supporting code under the web document root with an extension that is not specifically handled by the server.

### 详细描述:

如果代码存储在扩展名为“.inc”或“.pl”的文件中，并且Web服务器没有该扩展名的处理程序，则服务器可能会将该文件的内容直接发送给请求者没有预期的预处理。当该文件包含敏感信息（如数据库凭据）时，这可能允许攻击者破坏应用程序或关联的组件。

### Extended Description:

If code is stored in a file with an extension such as ".inc" or ".pl", and the web server does not have a handler for that extension, then the server will likely send the contents of the file directly to the requester without the pre-processing that was expected. When that file contains sensitive information such as database credentials, this may allow the attacker to compromise the application or associated components.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这与Web根（CWE-219）下的直接请求（CWE-425），备用路径（CWE-424），权限（CWE-275）和敏感文件重叠。

Weakness ID: 434

提交日期 2010-02-16---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Unrestricted Upload of File with Dangerous Type zh: -->无限制上传危险类型的文件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件允许攻击者上传或传输可在产品环境中自动处理的危险类型的文件。

### Description:

The software allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

当产品尝试时，这可能与不完整的黑名单/许可白名单错误具有链接关系，但是失败，以正确限制允许哪些类型的文件（CWE-183，CWE-184）。这也可能使中间人的多个解释错误重叠，例如：防病毒产品，不会删除或隔离可由客户端系统处理的某些文件扩展名的附件。

Weakness ID: 435

提交日期 2008-04-11---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Interaction Between Multiple Correctly-Behaving Entities zh: -->多个正确行为实体之间的不正确交互*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当两个实体在彼此独立运行时具有正确的行为时会发生交互错误，但是当它们作为组件集成到更大的系统或进程中时，它们会引入可能导致结果弱点的错误行为。

### Description:

An interaction error occurs when two entities have correct behavior when running independently of each other, but when they are integrated as components in a larger system or process, they introduce incorrect behaviors that may cause resultant weaknesses.

### 详细描述:

当系统或流程组合多个独立组件时，这通常会在系统级别产生新的紧急行为。但是，如果这些组成部分之间的相互作用没有得到充分考虑，那么一些紧急行为可能是不正确的，甚至是不安全的。

### Extended Description:

When a system or process combines multiple independent components, this often produces new, emergent behaviors at the system level. However, if the interactions between these components are not fully accounted for, some of the emergent behaviors can be incorrect or even insecure.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

CWE和其他地方的“交互错误”术语仅用于描述符合规范的产品。当一个或多个产品不符合规范时，则更可能是API滥用（CWE-227）或解释冲突（CWE-436）。在现实世界的场景中，这种区别可能会模糊不清，特别是当“事实上的”标准不符合规范时，或者没有标准但是广泛采用时。结果，在映射和分类期间可能难以区分这些弱点。

Weakness ID: 436

提交日期 2008-04-11---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Interpretation Conflict zh: -->解释冲突*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品A处理与产品B不同的输入或步骤，这导致A根据其对B状态的感知执行不正确的操作。

### Description:

Product A handles inputs or steps differently than Product B, which causes A to perform incorrect actions based on its perception of B's state.

### 详细描述:

这通常存在于代理，防火墙，防病毒软件和其他中间设备中，这些设备根据客户端或服务器的行为方式监控，允许，拒绝或修改流量。

### Extended Description:

This is generally found in proxies, firewalls, anti-virus software, and other intermediary devices that monitor, allow, deny, or modify traffic based on how the client or server is expected to behave.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 437

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incomplete Model of Endpoint Features zh: -->不完整的端点特征模型*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品充当两个或多个端点之间的中介或监视器，但它没有端点的功能，行为或状态的完整模型，可能导致产品基于此不完整的模型执行不正确的操作。

### Description:

A product acts as an intermediary or monitor between two or more endpoints, but it does not have a complete model of an endpoint's features, behaviors, or state, potentially causing the product to perform incorrect actions based on this incomplete model.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这可能与交互错误有关，尽管在某些情况下，其中一个端点根据规范无法正确执行。

Weakness ID: 439

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Behavioral Change in New Version or Environment zh: -->新版本或环境中的行为改变*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

A的行为或功能随着新版本的A或新环境而变化，而新环境由B未知（或可管理）。

### Description:

A's behavior or functionality changes with a new version of A, or a new environment, which is not known (or manageable) by B.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 440

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Expected Behavior Violation zh: -->预期的行为违规*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品使用的功能，API或功能的行为与产品预期的不同。

### Description:

A feature, API, or function being used by a product behaves differently than the product expects.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

有效性的一致性维度是预期行为违规的最合适的相关属性。也就是说，应用程序的行为与开发人员的期望不一致，导致违反软件的有效性属性。

Weakness ID: 441

提交日期 2013-02-21---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Unintended Proxy or Intermediary ('Confused Deputy') zh: -->非预期的代理人或中介人（“困惑的代理人”）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件从上游组件接收请求，消息或指令，但在将请求转发给软件控制范围之外的外部参与者之前，软件不能充分保留请求的原始源。这导致软件看起来是请求的来源，使其充当上游组件和外部参与者之间的代理或其他中介。

### Description:

The software receives a request, message, or directive from an upstream component, but the software does not sufficiently preserve the original source of the request before forwarding the request to an external actor that is outside of the software's control sphere. This causes the software to appear to be the source of the request, leading it to act as a proxy or other intermediary between the upstream component and the external actor.

### 详细描述:

如果攻击者无法直接联系目标，但软件可以访问目标，则攻击者可以向软件发送请求并将其从目标转发。该请求似乎来自软件系统，而不是攻击者的系统。因此，攻击者可以绕过访问控制（例如防火墙）或隐藏恶意请求的来源，因为请求不会直接来自攻击者。  
由于代理功能和邮件转发通常用于合法目的，因此在以下情况下此问题仅会成为漏洞：  
  
  
软件以不同的权限运行或在不同的系统上运行，或者具有与上游组件不同的访问级别;  
防止攻击者直接向目标发出请求;和  
攻击者可以创建代理未明确打算代表请求者转发的请求。此类请求可能指向意外的主机名，端口号或服务。或者，请求可能会发送到允许的服务，但请求可能包含不允许的指令，命令或资源。

### Extended Description:

If an attacker cannot directly contact a target, but the software has access to the target, then the attacker can send a request to the software and have it be forwarded from the target. The request would appear to be coming from the software's system, not the attacker's system. As a result, the attacker can bypass access controls (such as firewalls) or hide the source of malicious requests, since the requests would not be coming directly from the attacker.  
Since proxy functionality and message-forwarding often serve a legitimate purpose, this issue only becomes a vulnerability when:  
  
  
The software runs with different privileges or on a different system, or otherwise has different levels of access than the upstream component;  
The attacker is prevented from making the request directly to the target; and  
The attacker can create a request that the proxy does not explicitly intend to be forwarded on the behalf of the requester. Such a request might point to an unexpected hostname, port number, or service. Or, the request might be sent to an allowed service, but the request could contain disallowed directives, commands, or resources.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这个弱点与CWE-668（资源暴露于错误的球体）存在链接关系，因为代理有效地为攻击者提供了攻击者无法直接获取的目标资源的访问权限。

Weakness ID: 443

提交日期 I`m don`t know---> 修改日期 2009-10-29

* **Weakness Name:** *en: --> DEPRECATED (Duplicate): HTTP response splitting zh: -->DEPRECATED（Duplicate）：HTTP响应拆分*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这个弱点可以在CWE-113找到。

### Description:

This weakness can be found at CWE-113.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 444

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') zh: -->HTTP请求的不一致解释（'HTTP请求走私'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当用户和Web服务器（例如代理或防火墙）之间的数据流中的一个或多个实体解释格式错误或异常HTTP请求时，它们可能会被解释为不一致，从而允许攻击者将请求“偷偷”到一个设备没有其他设备意识到它。

### Description:

When malformed or abnormal HTTP requests are interpreted by one or more entities in the data flow between the user and the web server, such as a proxy or firewall, they can be interpreted inconsistently, allowing the attacker to "smuggle" a request to one device without the other device being aware of it.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

由于多重解释错误，目标是中间人或监视器，可以通过一致性操作（Transfer-Encoding和Content-Length头）执行请求走私。

Weakness ID: 446

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> UI Discrepancy for Security Feature zh: -->安全功能的UI差异*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

用户界面未正确启用或配置安全功能，但界面提供的反馈使用户相信该功能处于安全状态。

### Description:

The user interface does not correctly enable or configure a security feature, but the interface provides feedback that causes the user to believe that the feature is in a secure state.

### 详细描述:

当用户界面没有正确反映用户询问的内容时，它可能会导致用户产生错误的安全感。例如，用户可能会选中一个框以启用安全选项以启用加密通信，但该软件实际上并未启用加密。或者，用户可能提供“限制所有”访问控制规则，但该软件仅实现“限制某些”。

### Extended Description:

When the user interface does not properly reflect what the user asks of it, then it can lead the user into a false sense of security. For example, the user might check a box to enable a security option to enable encrypted communications, but the software does not actually enable the encryption. Alternately, the user might provide a "restrict ALL'" access control rule, but the software only implements "restrict SOME".

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这通常是结果。

Weakness ID: 447

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Unimplemented or Unsupported Feature in UI zh: -->UI中未实现或不支持的功能*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

似乎支持用于安全功能的UI功能，并向用户提供反馈，表明它受支持，但未实现基础功能。

### Description:

A UI function for a security feature appears to be supported and gives feedback to the user that suggests that it is supported, but the underlying functionality is not implemented.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这个问题需要更多的研究，因为没有太多的例子。目前尚不清楚它是主要还是结果。

Weakness ID: 448

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Obsolete Feature in UI zh: -->UI中的过时功能*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

UI功能已过时，产品不会警告用户。

### Description:

A UI function is obsolete and the product does not warn the user.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 449

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> The UI Performs the Wrong Action zh: -->UI执行错误的操作*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

UI针对用户的请求执行错误的操作。

### Description:

The UI performs the wrong action with respect to the user's request.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 450

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Multiple Interpretations of UI Input zh: -->UI输入的多重解释*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

UI具有用户输入的多种解释，但在选择较不安全的解释时不会提示用户。

### Description:

The UI has multiple interpretations of user input but does not prompt the user when it selects the less secure interpretation.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 451

提交日期 2014-02-18---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> User Interface (UI) Misrepresentation of Critical Information zh: -->用户界面（UI）错误表示关键信息*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

用户界面（UI）不能正确地向用户表示关键信息，从而使信息（或其来源）变得模糊或欺骗。这通常是网络钓鱼攻击的一个组成部分。

### Description:

The user interface (UI) does not properly represent critical information to the user, allowing the information - or its source - to be obscured or spoofed. This is often a component in phishing attacks.

### 详细描述:

如果攻击者可能导致UI显示错误数据，或者说服用户显示看似来自可信来源的信息，则攻击者可能欺骗用户执行错误操作。这通常是网络钓鱼攻击的一个组成部分，但存在其他类型的问题。例如，如果UI用于监视系统或网络的安全状态，则省略或模糊重要指示符可能会阻止用户检测并响应安全关键事件。  
用户界面虚假陈述可以采取多种形式：  
  
  
指示不正确：显示不正确的信息，这会阻止用户了解软件的真实状态或软件正在监控的环境，尤其是潜在危险的情况或操作。这可以分解为几种不同的子类型。  
叠加：显示区域用于提供关键信息，但另一个进程可以通过在显示屏上覆盖另一个元素来修改显示。用户没有与用户界面的预期部分交互。这是启用点击劫持攻击的问题，尽管存在涉及覆盖的许多其他类型的攻击。  
图标操作：错误的图标或错误的颜色指示符可能会受到影响（例如使危险的.EXE可执行文件看起来像无害的.GIF）  
时间安排：软件正在执行状态转换或上下文切换，并通过指示器呈现给用户，但竞争条件可能导致在产品完全切换上下文之前使用错误的指示符。如果攻击者可以触发错误，则可以无限期地延长竞赛窗口。  
可视截断：重要信息可能会从显示中截断，例如带有危险扩展名的长文件名，因为恶意部分被截断，因此未在GUI中显示。使用过多的空格也会导致截断，或将潜在危险的指示符放在用户视野之外（例如“filename.txt .exe”）。当由于长度以外的原因而移除信息的一部分时会发生不同类型的截断，例如在输入的中间意外插入输入结束标记，例如C中的NUL字节式字符串。  
视觉区分：视觉信息可能以一种使用户难以快速正确地区分显示器的关键和不重要部分的方式呈现。  
同形函数：来自不同字符集，字体或语言的字母可能看起来非常相似（即可能在视觉上等效），导致人类用户误读文本（例如，进行网络钓鱼攻击以欺骗用户访问恶意网站，其名称与可信站点名称相似。这可以被视为一种视觉区分问题。

### Extended Description:

If an attacker can cause the UI to display erroneous data, or to otherwise convince the user to display information that appears to come from a trusted source, then the attacker could trick the user into performing the wrong action. This is often a component in phishing attacks, but other kinds of problems exist. For example, if the UI is used to monitor the security state of a system or network, then omitting or obscuring an important indicator could prevent the user from detecting and reacting to a security-critical event.  
UI misrepresentation can take many forms:  
  
  
Incorrect indicator: incorrect information is displayed, which prevents the user from understanding the true state of the software or the environment the software is monitoring, especially of potentially-dangerous conditions or operations. This can be broken down into several different subtypes.  
Overlay: an area of the display is intended to give critical information, but another process can modify the display by overlaying another element on top of it. The user is not interacting with the expected portion of the user interface. This is the problem that enables clickjacking attacks, although many other types of attacks exist that involve overlay.  
Icon manipulation: the wrong icon, or the wrong color indicator, can be influenced (such as making a dangerous .EXE executable look like a harmless .GIF)  
Timing: the software is performing a state transition or context switch that is presented to the user with an indicator, but a race condition can cause the wrong indicator to be used before the product has fully switched context. The race window could be extended indefinitely if the attacker can trigger an error.  
Visual truncation: important information could be truncated from the display, such as a long filename with a dangerous extension that is not displayed in the GUI because the malicious portion is truncated. The use of excessive whitespace can also cause truncation, or place the potentially-dangerous indicator outside of the user's field of view (e.g. "filename.txt .exe"). A different type of truncation can occur when a portion of the information is removed due to reasons other than length, such as the accidental insertion of an end-of-input marker in the middle of an input, such as a NUL byte in a C-style string.  
Visual distinction: visual information might be presented in a way that makes it difficult for the user to quickly and correctly distinguish between critical and unimportant segments of the display.  
Homographs: letters from different character sets, fonts, or languages can appear very similar (i.e. may be visually equivalent) in a way that causes the human user to misread the text (for example, to conduct phishing attacks to trick a user into visiting a malicious web site with a visually-similar name as a trusted site). This can be regarded as a type of visual distinction issue.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目可以分解为较小的条目。它可能更像是一个类而不是一个基类。

Weakness ID: 453

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Insecure Default Variable Initialization zh: -->不安全的默认变量初始化*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

默认情况下，软件使用不安全或不太安全的值初始化内部变量。

### Description:

The software, by default, initializes an internal variable with an insecure or less secure value than is possible.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这与其他类别重叠，可能应该拆分为单独的项目。

Weakness ID: 454

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> External Initialization of Trusted Variables or Data Stores zh: -->可信变量或数据存储的外部初始化*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用可由不信任的参与者修改的输入初始化关键内部变量或数据存储。

### Description:

The software initializes critical internal variables or data stores using inputs that can be modified by untrusted actors.

### 详细描述:

软件系统应该不愿信任已在其信任边界之外初始化的变量，特别是如果它们由用户初始化。变量可能未正确初始化。如果攻击者可以初始化变量，那么他们可以影响易受攻击的系统将做什么。

### Extended Description:

A software system should be reluctant to trust variables that have been initialized outside of its trust boundary, especially if they are initialized by users. The variables may have been initialized incorrectly. If an attacker can initialize the variable, then they can influence what the vulnerable system will do.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

重叠缺少变量初始化，尤其是在PHP中。

Weakness ID: 455

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Non-exit on Failed Initialization zh: -->初始化失败时不退出*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在初始化期间发生与安全相关的错误时，软件不会退出或以其他方式修改其操作，例如当配置文件具有格式错误时，这可能导致软件以不如管理员预期的安全方式执行。

### Description:

The software does not exit or otherwise modify its operation when security-relevant errors occur during initialization, such as when a configuration file has a format error, which can cause the software to execute in a less secure fashion than intended by the administrator.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。这些问题并不经常报道，很难找到已发表的例子。

Weakness ID: 456

提交日期 2013-02-21---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Missing Initialization of a Variable zh: -->缺少变量的初始化*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会初始化关键变量，这会导致执行环境使用意外值。

### Description:

The software does not initialize critical variables, which causes the execution environment to use unexpected values.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这个弱点是导致许多弱点的主要因素，特别是在允许全局变量初始化（例如PHP）的Web应用程序中，可以直接请求库。

Weakness ID: 457

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Uninitialized Variable zh: -->使用未初始化的变量*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码使用尚未初始化的变量，从而导致不可预测或意外的结果。

### Description:

The code uses a variable that has not been initialized, leading to unpredictable or unintended results.

### 详细描述:

在某些语言（如C和C ++）中，默认情况下不会初始化堆栈变量。它们通常包含在调用函数之前具有堆栈内存的垃圾数据。攻击者有时可以控制或读取这些内容。在其他语言或条件中，未显式初始化的变量可以被赋予具有安全隐患的默认值，具体取决于程序的逻辑。存在未初始化的变量有时可能表示代码中存在排版错误。

### Extended Description:

In some languages such as C and C++, stack variables are not initialized by default. They generally contain junk data with the contents of stack memory before the function was invoked. An attacker can sometimes control or read these contents. In other languages or conditions, a variable that is not explicitly initialized can be given a default value that has security implications, depending on the logic of the program. The presence of an uninitialized variable can sometimes indicate a typographic error in the code.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 458

提交日期 2008-04-11---> 修改日期 2009-10-29

* **Weakness Name:** *en: --> DEPRECATED: Incorrect Initialization zh: -->DEPRECATED：初始化不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这个弱点已被弃用，因为它的名称和描述不匹配。描述重复CWE-454，而名称提示更抽象的初始化问题。有关更抽象的问题，请参阅CWE-665。

### Description:

This weakness has been deprecated because its name and description did not match. The description duplicated CWE-454, while the name suggested a more abstract initialization problem. Please refer to CWE-665 for the more abstract problem.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 459

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incomplete Cleanup zh: -->不完整的清理*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件在使用后没有正确“清理”并删除临时或支持资源。

### Description:

The software does not properly "clean up" and remove temporary or supporting resources after they have been used.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

CWE-459是CWE-404的孩子，因为虽然CWE-404涵盖任何类型的不正确关闭或资源释放，但CWE-459专门处理多步骤关闭过程，其中“正确”清理的关键步骤被省略或不可能。也就是说，CWE-459专门处理无法成功删除所有潜在敏感数据的清理或关闭过程。

Weakness ID: 460

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Cleanup on Thrown Exception zh: -->抛出异常的不正确清理*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当抛出异常时，产品不会清除其状态或错误地清除其状态，从而导致意外状态或控制流。

### Description:

The product does not clean up its state or incorrectly cleans up its state when an exception is thrown, leading to unexpected state or control flow.

### 详细描述:

通常，当函数或循环变得复杂时，在整个执行过程中需要一定程度的资源清理。例外可能会扰乱代码流并阻止必要的清理。

### Extended Description:

Often, when functions or loops become complicated, some level of resource cleanup is needed throughout execution. Exceptions can disturb the flow of the code and prevent the necessary cleanup from happening.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 462

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Duplicate Key in Associative List (Alist) zh: -->关联列表中的重复键（Alist）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

关联列表中的重复键可能导致非唯一键被误认为是错误。

### Description:

Duplicate keys in associative lists can lead to non-unique keys being mistaken for an error.

### 详细描述:

重复键输入 - 如果alist设计正确 - 可以用作常量时间替换功能。但是，可能会错误地插入重复的键条目。由于这种歧义，不建议在关联列表中使用重复的键条目，不应该允许。

### Extended Description:

A duplicate key entry -- if the alist is designed properly -- could be used as a constant time replace function. However, duplicate key entries could be inserted by mistake. Because of this ambiguity, duplicate key entries in an association list are not recommended and should not be allowed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 463

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Deletion of Data Structure Sentinel zh: -->删除数据结构Sentinel*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

意外删除数据结构标记可能会导致严重的编程逻辑问题。

### Description:

The accidental deletion of a data-structure sentinel can cause serious programming logic problems.

### 详细描述:

通常，数据结构标记用于标记数据结构的结构。一个常见的例子是字符串末尾的空字符。另一个常见的例子是链表，它可能包含一个标记列表末尾的标记。允许这种类型的控制数据易于访问是危险的。因此，重要的是防止在提供安全性的某些包装器接口之外的删除或修改。

### Extended Description:

Often times data-structure sentinels are used to mark structure of the data structure. A common example of this is the null character at the end of strings. Another common example is linked lists which may contain a sentinel to mark the end of the list. It is dangerous to allow this type of control data to be easily accessible. Therefore, it is important to protect from the deletion or modification outside of some wrapper interface which provides safety.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 464

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Addition of Data Structure Sentinel zh: -->添加数据结构Sentinel*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

意外添加数据结构标记会导致严重的编程逻辑问题。

### Description:

The accidental addition of a data-structure sentinel can cause serious programming logic problems.

### 详细描述:

数据结构标记通常用于标记数据结构。一个常见的例子是字符串末尾的空字符或标记链表结尾的特殊标记。允许这种类型的控制数据易于访问是危险的。因此，保护​​免受哨兵的添加或修改是很重要的。

### Extended Description:

Data-structure sentinels are often used to mark the structure of data. A common example of this is the null character at the end of strings or a special sentinel to mark the end of a linked list. It is dangerous to allow this type of control data to be easily accessible. Therefore, it is important to protect from the addition or modification of sentinels.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 466

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Return of Pointer Value Outside of Expected Range zh: -->超出预期范围的指针值的返回*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

函数可以返回指向指针应该引用的缓冲区之外的内存的指针。

### Description:

A function can return a pointer to memory that is outside of the buffer that the pointer is expected to reference.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目应与CWE-119具有链接关系，而不是父/子关系，但是此弱点的焦点并未完全映射到CWE中的任何现有条目。正在考虑一个新的父级，它涵盖了不正确的返回值的更普遍的问题。还有一个与弱点的抽象关系，其中一个组件向另一个组件发送不正确的消息;在这种情况下，一个例程正在向另一个例程发送不正确的值。

Weakness ID: 467

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of sizeof() on a Pointer Type zh: -->在指针类型上使用sizeof（）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码在malloced指针类型上调用sizeof（），它总是返回wordsize / 8。如果程序员打算确定已分配了多少内存，则会产生意外结果。

### Description:

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

### 详细描述:

在指针上使用sizeof（）有时可以生成有用的信息。一个明显的例子是在平台上找出单词大小。通常，sizeof（指针）的外观表示错误。

### Extended Description:

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 468

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Pointer Scaling zh: -->指针缩放不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在C和C ++中，由于数学运算被隐式缩放的语义，人们可能经常意外地引用错误的内存。

### Description:

In C and C++, one may often accidentally refer to the wrong memory due to the semantics of when math operations are implicitly scaled.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 469

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Pointer Subtraction to Determine Size zh: -->使用指针减法来确定大小*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序从另一个指针中减去一个指针以确定大小，但如果指针不存在于同一个内存块中，则此计算可能不正确。

### Description:

The application subtracts one pointer from another in order to determine size, but this calculation can be incorrect if the pointers do not exist in the same memory chunk.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 470

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection') zh: -->使用外部控制输入来选择类或代码（'不安全反射'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序使用带反射的外部输入来选择要使用的类或代码，但它不足以防止输入选择不正确的类或代码。

### Description:

The application uses external input with reflection to select which classes or code to use, but it does not sufficiently prevent the input from selecting improper classes or code.

### 详细描述:

如果应用程序使用外部输入来确定要实例化的类或要调用的方法，则攻击者可以提供值以选择意外的类或方法。如果发生这种情况，则攻击者可以创建开发人员不想要的控制流路径。这些路径可能会绕过身份验证或访问控制检查，或以其他方式导致应用程序以意外方式运行。如果攻击者可以将文件上载到应用程序的类路径（CWE-427）上显示的位置，或者将新条目添加到应用程序的类路径（CWE-426），则这种情况将成为世界末日。在这些条件中的任何一种情况下，攻击者都可以使用反射将新的恶意行为引入应用程序。

### Extended Description:

If the application uses external inputs to determine which class to instantiate or which method to invoke, then an attacker could supply values to select unexpected classes or methods. If this occurs, then the attacker could create control flow paths that were not intended by the developer. These paths could bypass authentication or access control checks, or otherwise cause the application to behave in an unexpected manner. This situation becomes a doomsday scenario if the attacker can upload files into a location that appears on the application's classpath (CWE-427) or add new entries to the application's classpath (CWE-426). Under either of these conditions, the attacker can use reflection to introduce new, malicious behavior into the application.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 471

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Modification of Assumed-Immutable Data (MAID) zh: -->假定不可变数据的修改（MAID）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法正确保护假定不可变元素不被攻击者修改。

### Description:

The software does not properly protect an assumed-immutable element from being modified by an attacker.

### 详细描述:

当一个特定的输入对于应用程序的运行至关重要而它根本不应该是可修改的时候就会发生这种情况，但确实如此。某些资源通常被认为是不可变的，例如Web应用程序中的隐藏表单字段，cookie和反向DNS查找。

### Extended Description:

This occurs when a particular input is critical enough to the functioning of the application that it should not be modifiable at all, but it is. Certain resources are often assumed to be immutable when they are not, such as hidden form fields in web applications, cookies, and reverse DNS lookups.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

MAID问题可能是许多其他弱点的主要问题，它们是可以轻松访问内部程序结构的语言的主要因素，例如PHP的register\_globals和类似功能。但是，MAID问题也可能是由修改内部状态的弱点引起的;例如，程序可能会验证某些数据并将其存储在内存中，但缓冲区溢出可能会覆盖该验证数据，从而导致程序逻辑发生变化。

Weakness ID: 472

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> External Control of Assumed-Immutable Web Parameter zh: -->假设不可变Web参数的外部控制*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序不能充分验证假定为不可变但实际上是外部可控的输入，例如隐藏的表单字段。

### Description:

The web application does not sufficiently verify inputs that are assumed to be immutable but are actually externally controllable, such as hidden form fields.

### 详细描述:

如果Web产品未正确保护隐藏表单字段，参数，cookie或URL中的假定不可变值，则可能导致对关键数据的修改。 Web应用程序经常错误地假设在隐藏字段或cookie中传递给客户端的数据不容易被篡改。对用户可控制的数据进行不正确的验证可能会导致应用程序处理不正确，通常是恶意的输入。  
例如，自定义cookie通常跨会话存储会话数据或持久数据。这种会话数据通常涉及服务器端的安全相关决策，例如用户认证和访问控制。因此，cookie可能包含敏感数据，例如用户凭据和权限。这是一种危险的做法，因为它通常会导致服务器端应用程序不正确地依赖客户端提供的cookie的价值。

### Extended Description:

If a web product does not properly protect assumed-immutable values from modification in hidden form fields, parameters, cookies, or URLs, this can lead to modification of critical data. Web applications often mistakenly make the assumption that data passed to the client in hidden fields or cookies is not susceptible to tampering. Improper validation of data that are user-controllable can lead to the application processing incorrect, and often malicious, input.  
For example, custom cookies commonly store session data or persistent data across sessions. This kind of session data is normally involved in security related decisions on the server side, such as user authentication and access control. Thus, the cookies might contain sensitive data such as user credentials and privileges. This is a dangerous practice, as it can often lead to improper reliance on the value of the client-provided cookie by the server side application.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这是许多其他弱点和功能后果的主要弱点，包括XSS，SQL注入，路径公开和文件包含。

Weakness ID: 473

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> PHP External Variable Modification zh: -->PHP外部变量修改*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

PHP应用程序无法正确防止从外部源修改变量，例如查询参数或cookie。这可能会使应用程序面临许多其他方面不存在的弱点。

### Description:

A PHP application does not properly protect against the modification of variables from external sources, such as query parameters or cookies. This can expose the application to numerous weaknesses that would not exist otherwise.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这是特定语言的假定不可变数据修改（MAID）实例。这可能是直接请求（备用路径）问题的结果。它可能是PHP文件包含，SQL注入，XSS，身份验证绕过等弱点的主要原因。

Weakness ID: 474

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Function with Inconsistent Implementations zh: -->函数与不一致实现的使用*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码使用的函数跨操作系统和版本具有不一致的实现。

### Description:

The code uses a function that has inconsistent implementations across operating systems and versions.

### 详细描述:

当代码在程序员期望的不同环境下移植或构建时，使用不一致的实现可能会导致行为更改，这可能会在某些情况下导致安全问题。  
许多功能的实现因平台而异，有时甚至是同一平台的不同版本。实施差异包括：  
  
  
解释参数的方式略有不同会导致结果不一致。  
该功能的某些实现具有重大的安全风险。  
可能未在所有平台上定义该功能。  
该函数可能会更改它可以提供的返回码，或更改其返回码的含义。

### Extended Description:

The use of inconsistent implementations can cause changes in behavior when the code is ported or built under a different environment than the programmer expects, which can lead to security problems in some cases.  
The implementation of many functions varies by platform, and at times, even by different versions of the same platform. Implementation differences can include:  
  
  
Slight differences in the way parameters are interpreted leading to inconsistent results.  
Some implementations of the function carry significant security risks.  
The function might not be defined on all platforms.  
The function might change which return codes it can provide, or change the meaning of its return codes.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 475

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Undefined Behavior for Input to API zh: -->API输入的未定义行为*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

除非将其控制参数设置为特定值，否则此函数的行为是未定义的。

### Description:

The behavior of this function is undefined unless its control parameter is set to a specific value.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

libc的Linux标准基础规范2.0.1对一些内部函数的参数设置了约束[21]。如果不满足约束，则不定义函数的行为。直接调用此函数是不常见的。它几乎总是通过系统头文件中定义的宏调用，并且宏确保满足以下约束：值1必须传递给以下文件系统函数的第三个参数（版本号）：\_\_ xmknod必须将值2传递给以下宽字符串函数的第三个参数（组参数）：\_\_ wcstod\_internal \_\_wcstof\_internal \_\_wcstol\_internal \_\_wcstold\_internal \_\_wcstoul\_internal值3必须作为以下文件系统函数的第一个参数（版本号）传递：\_\_ xstat \_\_lxstat \_\_fxstat \_\_xstat64 \_\_lxstat64 \_\_fxstat64

Weakness ID: 476

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> NULL Pointer Dereference zh: -->NULL指针解除引用*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当应用程序取消引用它期望有效的指针但是为NULL时，会发生NULL指针解除引用，通常会导致崩溃或退出。

### Description:

A NULL pointer dereference occurs when the application dereferences a pointer that it expects to be valid, but is NULL, typically causing a crash or exit.

### 详细描述:

NULL指针解引用问题可能通过许多缺陷发生，包括竞争条件和简单的编程遗漏。

### Extended Description:

NULL pointer dereference issues can occur through a number of flaws, including race conditions, and simple programming omissions.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 477

提交日期 2008-01-30---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Obsolete Function zh: -->使用过时功能*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码使用已弃用或过时的函数，这表明代码尚未被主动审查或维护。

### Description:

The code uses deprecated or obsolete functions, which suggests that the code has not been actively reviewed or maintained.

### 详细描述:

随着编程语言的发展，由于以下原因，函数偶尔会变得过时：  
  
  
语言的进步  
更好地理解如何有效和安全地执行操作  
管理某些操作的约定的变化  
  
  
被删除的函数通常被更新的对应函数替换，这些对应函数以某种不同的并且希望改进的方式执行相同的任务。

### Extended Description:

As programming languages evolve, functions occasionally become obsolete due to:  
  
  
Advances in the language  
Improved understanding of how operations should be performed effectively and securely  
Changes in the conventions that govern certain operations  
  
  
Functions that are removed are usually replaced by newer counterparts that perform the same task in some different and hopefully improved way.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 478

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Missing Default Case in Switch Statement zh: -->缺少Switch语句中的默认情况*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码在switch语句中没有默认情况，这可能导致复杂的逻辑错误和由此产生的弱点。

### Description:

The code does not have a default case in a switch statement, which might lead to complex logical errors and resultant weaknesses.

### 详细描述:

该缺陷代表了软件开发中的常见问题，其中并非所有可能的变量值都由给定过程考虑或处理。因此，根据不良信息和级联故障结果做出进一步的决策。这种级联故障可能导致任何数量的安全问题，并构成系统中的重大故障。

### Extended Description:

This flaw represents a common problem in software development, in which not all possible values for a variable are considered or handled by a given process. Because of this, further decisions are made based on poor information, and cascading failure results. This cascading failure may result in any number of security issues, and constitutes a significant failure in the system.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 479

提交日期 2010-12-13---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Signal Handler Use of a Non-reentrant Function zh: -->信号处理程序使用非重入函数*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序定义了一个调用非重入函数的信号处理程序。

### Description:

The program defines a signal handler that calls a non-reentrant function.

### 详细描述:

非重入函数是在第一次调用完成之前无法安全地调用，中断和调用的函数，而不会导致内存损坏。这可能导致意外的系统状态具有不可预测的结果，并且具有取决于上下文的各种潜在后果，包括拒绝服务和代码执行。  
许多函数不是可重入的，但如果在信号处理程序中使用它们中的一些可能会导致内存损坏。函数调用syslog（）就是一个例子。为了执行其功能，它将少量内存分配为“临时空间”。如果syslog（）被信号调用挂起并且信号处理程序调用syslog（），则这两个函数使用的内存将进入未定义且可能可利用的状态。 malloc（）和free（）的实现管理全局结构中的元数据，以便跟踪分配哪个内存与哪个内存可用，但它们是不可重入的。同时调用这些函数可能会导致元数据损坏。

### Extended Description:

Non-reentrant functions are functions that cannot safely be called, interrupted, and then recalled before the first call has finished without resulting in memory corruption. This can lead to an unexpected system state an unpredictable results with a variety of potential consequences depending on context, including denial of service and code execution.  
Many functions are not reentrant, but some of them can result in the corruption of memory if they are used in a signal handler. The function call syslog() is an example of this. In order to perform its functionality, it allocates a small amount of memory as "scratch space." If syslog() is suspended by a signal call and the signal handler calls syslog(), the memory used by both of these functions enters an undefined, and possibly, exploitable state. Implementations of malloc() and free() manage metadata in global structures in order to track which memory is allocated versus which memory is available, but they are non-reentrant. Simultaneous calls to these functions can cause corruption of the metadata.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 480

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Incorrect Operator zh: -->使用不正确的操作员*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

程序员意外地使用了错误的操作符，它以安全相关的方式更改应用程序逻辑。

### Description:

The programmer accidentally uses the wrong operator, which changes the application logic in security-relevant ways.

### 详细描述:

这些类型的错误通常是拼写错误的结果。

### Extended Description:

These types of errors are generally the result of a typo.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 481

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Assigning instead of Comparing zh: -->分配而不是比较*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当意图进行比较时，代码使用运算符进行赋值。

### Description:

The code uses an operator for assignment when the intention was to perform a comparison.

### 详细描述:

在许多语言中，compare语句在外观上与赋值语句非常接近，并且经常被混淆。此错误通常是拼写错误的结果，通常会导致程序执行出现明显问题。如果比较在if语句中，则if语句通常会评估谓词右侧的值。

### Extended Description:

In many languages the compare statement is very close in appearance to the assignment statement and are often confused. This bug is generally the result of a typo and usually causes obvious problems with program execution. If the comparison is in an if statement, the if statement will usually evaluate the value of the right-hand side of the predicate.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 482

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Comparing instead of Assigning zh: -->比较而不是分配*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在意图执行分配时，代码使用运算符进行比较。

### Description:

The code uses an operator for comparison when the intention was to perform an assignment.

### 详细描述:

在许多语言中，compare语句与赋值语句的外观非常接近;他们经常感到困惑。

### Extended Description:

In many languages, the compare statement is very close in appearance to the assignment statement; they are often confused.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 483

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Block Delimitation zh: -->块定界不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码未明确分隔旨在包含2个或更多语句的块，从而产生逻辑错误。

### Description:

The code does not explicitly delimit a block that is intended to contain 2 or more statements, creating a logic error.

### 详细描述:

在某些语言中，大括号（或其他分隔符）对于块是可选的。省略分隔符时，可以插入一个逻辑错误，其中一个语句被认为是在一个块中但不是。在某些情况下，逻辑错误可能会产生安全隐患。

### Extended Description:

In some languages, braces (or other delimiters) are optional for blocks. When the delimiter is omitted, it is possible to insert a logic error in which a statement is thought to be in a block but is not. In some cases, the logic error can have security implications.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 484

提交日期 2008-11-24---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Omitted Break Statement in Switch zh: -->Switch中省略的Break语句*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序在开关或类似构造中省略了break语句，导致与多个条件相关联的代码被执行。当程序员只打算执行与一个条件相关的代码时，这可能会导致问题。

### Description:

The program omits a break statement within a switch or similar construct, causing code associated with multiple conditions to execute. This can cause problems when the programmer only intended to execute code associated with one condition.

### 详细描述:

这可能导致关键代码在不应该执行的情况下执行。

### Extended Description:

This can lead to critical code executing in situations where it should not.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 486

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Comparison of Classes by Name zh: -->按名称比较类*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序按名称比较类，当多个类具有相同的名称时，可能导致它使用错误的类。

### Description:

The program compares classes by name, which can cause it to use the wrong class when multiple classes can have the same name.

### 详细描述:

如果决定信任对象的方法和数据是基于类的名称，则恶意用户可能发送与受信任类同名的对象，从而获得对已知类和类型的信任。

### Extended Description:

If the decision to trust the methods and data of an object is based on the name of a class, it is possible for malicious users to send objects of the same name as trusted classes and thereby gain the trust afforded to known classes and types.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 487

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Reliance on Package-level Scope zh: -->依赖于包级范围*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Java包本身并不是封闭的;因此，依靠它们来实现代码安全并不是一种好的做法。

### Description:

Java packages are not inherently closed; therefore, relying on them for code security is not a good practice.

### 详细描述:

包范围的目的是防止程序的其他部分意外访问。这是一种易于软件开发的功能，但不是安全功能。

### Extended Description:

The purpose of package scope is to prevent accidental access by other parts of a program. This is an ease-of-software-development feature but not a security feature.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 488

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Exposure of Data Element to Wrong Session zh: -->数据元素暴露于错误的会话*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品不足以强制执行不同会话状态之间的边界，从而导致数据被提供给错误会话或由错误会话使用。

### Description:

The product does not sufficiently enforce boundaries between the states of different sessions, causing data to be provided to, or used by, the wrong session.

### 详细描述:

数据可以通过单个对象的成员变量（例如Servlet）和来自共享池的对象从一个会话“流出”到另一个会话。  
对于Servlet，开发人员有时不明白，除非Servlet实现SingleThreadModel接口，否则Servlet是单例; Servlet只有一个实例，并且使用该实例并重新使用它来处理由不同线程同时处理的多个请求。一个常见的结果是开发人员使用Servlet成员字段，使得一个用户可能无意中看到另一个用户的数据。换句话说，在Servlet成员字段中存储用户数据会引入数据访问竞争条件。

### Extended Description:

Data can "bleed" from one session to another through member variables of singleton objects, such as Servlets, and objects from a shared pool.  
In the case of Servlets, developers sometimes do not understand that, unless a Servlet implements the SingleThreadModel interface, the Servlet is a singleton; there is only one instance of the Servlet, and that single instance is used and re-used to handle multiple requests that are processed simultaneously by different threads. A common result is that developers use Servlet member fields in such a way that one user may inadvertently see another user's data. In other words, storing user data in Servlet member fields introduces a data access race condition.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 489

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Leftover Debug Code zh: -->剩余的调试代码*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

可以使用可以创建非预期入口点的活动调试代码来部署应用程序。

### Description:

The application can be deployed with active debugging code that can create unintended entry points.

### 详细描述:

一种常见的开发实践是添加专门为调试或测试目的而设计的“后门”代码，该代码不打算随应用程序一起发布或部署。这些后门入口点会产生安全风险，因为它们在设计或测试期间不予考虑，并且超出了应用程序的预期运行条件。

### Extended Description:

A common development practice is to add "back door" code specifically designed for debugging or testing purposes that is not intended to be shipped or deployed with the application. These back door entry points create security risks because they are not considered during design or testing and fall outside of the expected operating conditions of the application.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在J2EE中，主要方法可能是调试代码留在应用程序中的一个很好的指示器，尽管可能没有任何直接的安全影响。

Weakness ID: 491

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Public cloneable() Method Without Final ('Object Hijack') zh: -->没有Final的公共cloneable（）方法（'Object Hijack'）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

类具有一个未声明为final的cloneable（）方法，它允许在不调用构造函数的情况下创建对象。这可能导致对象处于意外状态。

### Description:

A class has a cloneable() method that is not declared final, which allows an object to be created without calling the constructor. This can cause the object to be in an unexpected state.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 492

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Inner Class Containing Sensitive Data zh: -->使用包含敏感数据的内部类*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

内部类被转换为可在包范围内访问的类，并且可能暴露程序员打算对攻击者保密的代码。

### Description:

Inner classes are translated into classes that are accessible at package scope and may expose code that the programmer intended to keep private to attackers.

### 详细描述:

内部类因其转换为Java字节码的方式而悄然引入了一些安全问题。在Java源代码中，似乎可以声明内部类只能由封闭类访问，但Java字节码没有内部类的概念，因此编译器必须将内部类声明转换为具有包级别的对等类访问原始的外部类。更隐蔽的是，由于内部类可以访问其封闭类中的私有字段，因此一旦内部类成为字节码中的对等类，编译器就会将内部类访问的私有字段转换为受保护字段。

### Extended Description:

Inner classes quietly introduce several security concerns because of the way they are translated into Java bytecode. In Java source code, it appears that an inner class can be declared to be accessible only by the enclosing class, but Java bytecode has no concept of an inner class, so the compiler must transform an inner class declaration into a peer class with package level access to the original outer class. More insidiously, since an inner class can access private fields in their enclosing class, once an inner class becomes a peer class in bytecode, the compiler converts private fields accessed by the inner class into protected fields.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

移动代码（在本例中为Java Applet）是通过网络传输并在远程计算机上执行的代码。由于移动代码开发人员几乎无法控制其代码执行的环境，因此特殊的安全问题变得相关。最大的环境威胁之一是移动代码与其他可能是恶意的移动代码并行运行的风险。因为所有流行的Web浏览器都在同一个JVM中一起执行来自多个源的代码，所以移动代码的许多安全准则都集中在防止有权访问同一虚拟机的对手操纵对象的状态和行为。程序正在运行。

Weakness ID: 493

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Critical Public Variable Without Final Modifier zh: -->没有最终修饰符的关键公共变量*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品有一个非final的关键公共变量，它允许修改变量以包含意外值。

### Description:

The product has a critical public variable that is not final, which allows the variable to be modified to contain unexpected values.

### 详细描述:

如果某个字段是非final和public，则可以通过任何有权访问包含该字段的类的函数设置该值来更改该字段。如果程序的其他部分对该字段的内容做出假设，则可能导致漏洞。

### Extended Description:

If a field is non-final and public, it can be changed once the value is set by any function that has access to the class which contains the field. This could lead to a vulnerability if other parts of the program make assumptions about the contents of that field.

### 问题背景 (Background Detail):

移动代码（例如Java Applet）是通过网络传输并在远程机器上执行的代码。由于移动代码开发人员几乎无法控制其代码执行的环境，因此特殊的安全问题变得相关。最大的环境威胁之一是移动代码与其他可能是恶意的移动代码并行运行的风险。因为所有流行的Web浏览器都在同一个JVM中一起执行来自多个源的代码，所以移动代码的许多安全准则都集中在防止有权访问同一虚拟机的对手操纵对象的状态和行为。程序正在运行。

### 笔记 (Notes):

没有笔记

Weakness ID: 494

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Download of Code Without Integrity Check zh: -->下载没有完整性检查的代码*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品从远程位置下载源代码或可执行文件，并在不充分验证代码的来源和完整性的情况下执行代码。

### Description:

The product downloads source code or an executable from a remote location and executes the code without sufficiently verifying the origin and integrity of the code.

### 详细描述:

攻击者可以通过破坏主机服务器，执行DNS欺骗或修改传输中的代码来执行恶意代码。

### Extended Description:

An attacker can execute malicious code by compromising the host server, performing DNS spoofing, or modifying the code in transit.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这对于移动代码至关重要，但随着开发人员继续采用基于网络的自动化产品分发和升级，它可能会变得越来越普遍。软件即服务（SaaS）可能会带来额外的细微之处。常见的开发方案可能包括广告服务器妥协和不良升级。

Weakness ID: 495

提交日期 2019-01-03---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Private Data Structure Returned From A Public Method zh: -->从公共方法返回的私有数据结构*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品有一个声明为public的方法，但返回对私有数据结构的引用，然后可以以意外的方式对其进行修改。

### Description:

The product has a method that is declared public, but returns a reference to a private data structure, which could then be modified in unexpected ways.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 496

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Public Data Assigned to Private Array-Typed Field zh: -->公共数据分配给私有阵列类型字段*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

将公共数据分配给专用阵列等同于为该阵列提供公共访问权限。

### Description:

Assigning public data to a private array is equivalent to giving public access to the array.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 497

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Exposure of System Data to an Unauthorized Control Sphere zh: -->将系统数据暴露给未经授权的控制球体*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

公开系统数据或调试信息有助于对手了解系统并形成攻击计划。

### Description:

Exposing system data or debugging information helps an adversary learn about the system and form an attack plan.

### 详细描述:

当系统数据或调试信息通过输出流或记录功能离开程序时，会发生信息泄露，使未经授权的各方可以访问它。攻击者还可以通过向Web应用程序提交异常请求来导致错误。对这些错误的响应可以揭示详细的系统信息，拒绝服务，导致安全机制失败，以及使服务器崩溃。攻击者可以使用揭示技术，操作系统和产品版本的错误消息来调整针对这些技术中已知漏洞的攻击。应用程序可以使用提供重要实现细节的诊断方法，例如堆栈跟踪，作为其错误处理机制的一部分。

### Extended Description:

An information exposure occurs when system data or debugging information leaves the program through an output stream or logging function that makes it accessible to unauthorized parties. An attacker can also cause errors to occur by submitting unusual requests to the web application. The response to these errors can reveal detailed system information, deny service, cause security mechanisms to fail, and crash the server. An attacker can use error messages that reveal technologies, operating systems, and product versions to tune the attack against known vulnerabilities in these technologies. An application may use diagnostic methods that provide significant implementation details such as stack traces as part of its error handling mechanism.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 498

提交日期 2011-03-29---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Cloneable Class Containing Sensitive Information zh: -->包含敏感信息的可克隆类*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码包含一个包含敏感数据的类，但该类是可复制的。然后可以通过克隆类来访问数据。

### Description:

The code contains a class with sensitive data, but the class is cloneable. The data can then be accessed by cloning the class.

### 详细描述:

可克隆类是有效的开放类，因为数据不能隐藏在它们中。未明确拒绝克隆的类可以由任何其他类克隆，而无需运行构造函数。

### Extended Description:

Cloneable classes are effectively open classes, since data cannot be hidden in them. Classes that do not explicitly deny cloning can be cloned by any other class without running the constructor.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 499

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Serializable Class Containing Sensitive Data zh: -->包含敏感数据的可序列化类*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码包含一个包含敏感数据的类，但该类未明确拒绝序列化。可以通过另一个类序列化类来访问数据。

### Description:

The code contains a class with sensitive data, but the class does not explicitly deny serialization. The data can be accessed by serializing the class through another class.

### 详细描述:

可序列化类是有效的开放类，因为数据不能隐藏在它们中。未明确拒绝序列化的类可以由任何其他类序列化，然后可以依次使用存储在其中的数据。

### Extended Description:

Serializable classes are effectively open classes since data cannot be hidden in them. Classes that do not explicitly deny serialization can be serialized by any other class, which can then in turn use the data stored inside it.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 500

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Public Static Field Not Marked Final zh: -->公共静态字段未标记为最终*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

对象包含未标记为final的公共静态字段，这可能允许以意外方式对其进行修改。

### Description:

An object contains a public static field that is not marked final, which might allow it to be modified in unexpected ways.

### 详细描述:

公共静态变量可以在没有访问器的情况下读取，并且在没有应用程序中的任何类的mutator的情况下进行更改

### Extended Description:

Public static variables can be read without an accessor and changed without a mutator by any classes in the application.

### 问题背景 (Background Detail):

当一个字段被声明为public而不是final时，该字段可以被任意Java代码读取和写入。

### 笔记 (Notes):

没有笔记

Weakness ID: 501

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Trust Boundary Violation zh: -->信任边界违规*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品在相同的数据结构或结构化消息中混合可信和不可信的数据。

### Description:

The product mixes trusted and untrusted data in the same data structure or structured message.

### 详细描述:

信任边界可以被认为是通过程序绘制的线。在该行的一侧，数据是不可信的。在该行的另一侧，数据被认为是值得信赖的。验证逻辑的目的是允许数据安全地跨越信任边界 - 从不受信任转移到受信任。当程序模糊信任信任和不信任信号之间的界线时，就会发生信任边界违规。通过在相同的数据结构中组合可信和不可信的数据，程序员更容易错误地信任未经验证的数据。

### Extended Description:

A trust boundary can be thought of as line drawn through a program. On one side of the line, data is untrusted. On the other side of the line, data is assumed to be trustworthy. The purpose of validation logic is to allow data to safely cross the trust boundary - to move from untrusted to trusted. A trust boundary violation occurs when a program blurs the line between what is trusted and what is untrusted. By combining trusted and untrusted data in the same data structure, it becomes easier for programmers to mistakenly trust unvalidated data.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 502

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Deserialization of Untrusted Data zh: -->不受信任数据的反序列化*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序对不受信任的数据进行反序列化，但未充分验证结果数据是否有效。

### Description:

The application deserializes untrusted data without sufficiently verifying that the resulting data will be valid.

### 详细描述:

序列化对象进行通信或保存以供以后使用通常很方便。但是，如果不使用加密技术来保护自身，则通常可以在不使用提供的访问器功能的情况下修改反序列化的数据或代码。此外，任何加密仍然是客户端安全 - 这是一个危险的安全假设。  
不信任的数据不能被信任为格式良好。  
当开发人员对“小工具链”或者在反序列化过程中可以自行执行的一系列实例和方法调用（即，在对象返回给调用者之前）没有限制时，攻击者有时可以利用它们来实现执行未经授权的操作，例如生成shell。

### Extended Description:

It is often convenient to serialize objects for communication or to save them for later use. However, deserialized data or code can often be modified without using the provided accessor functions if it does not use cryptography to protect itself. Furthermore, any cryptography would still be client-side security -- which is a dangerous security assumption.  
Data that is untrusted can not be trusted to be well-formed.  
When developers place no restrictions on "gadget chains," or series of instances and method invocations that can self-execute during the deserialization process (i.e., before the object is returned to the caller), it is sometimes possible for attackers to leverage them to perform unauthorized actions, like generating a shell.

### 问题背景 (Background Detail):

序列化和反序列化是指获取程序内部对象相关数据的过程，以允许数据外部存储或传输（“序列化”）的方式打包，然后提取序列化数据以重建原始对象（“反序列化“）。

### 笔记 (Notes):

CWE-502和CWE-915之间的关系需要进一步探索。 CWE-915的范围更窄，适用于对象修改，不一定用于反序列化。

Weakness ID: 506

提交日期 2008-01-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Embedded Malicious Code zh: -->嵌入式恶意代码*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该应用程序包含看似恶意的代码。

### Description:

The application contains code that appears to be malicious in nature.

### 详细描述:

恶意瑕疵已经获得了色彩缤纷的名字，包括特洛伊木马，活板门，定时炸弹和逻辑炸弹。开发人员可能会插入恶意代码，以便在将来的某个时间破坏应用程序或其主机系统的安全性。它通常是指执行有用服务但以用户不想要的方式利用程序用户权限的程序。

### Extended Description:

Malicious flaws have acquired colorful names, including Trojan horse, trapdoor, timebomb, and logic-bomb. A developer might insert malicious code with the intent to subvert the security of an application or its host system at some time in the future. It generally refers to a program that performs a useful service but exploits rights of the program's user in a way the user does not intend.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

丹·爱德华兹（Dan Edwards）引入了“特洛伊木马”一词，由詹姆斯·安德森（James Anderson）[18]记录，描述了特定的计算机安全威胁。它被重新定义了很多次[4,18-20]。

Weakness ID: 507

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Trojan Horse zh: -->特洛伊木马*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件似乎包含良性或有用的功能，但它还包含违反正常操作的代码，这些代码违反了用户或系统管理员的预期安全策略。

### Description:

The software appears to contain benign or useful functionality, but it also contains code that is hidden from normal operation that violates the intended security policy of the user or the system administrator.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在运行时编译的潜在恶意动态代码可以隐藏不会出现在基线中的任何数量的攻击。使用动态编译的代码还可以允许对部署后的应用程序进行攻击。

Weakness ID: 508

提交日期 2008-01-30---> 修改日期 2012-10-30

* **Weakness Name:** *en: --> Non-Replicating Malicious Code zh: -->非复制恶意代码*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

非复制恶意代码仅驻留在受攻击的目标系统或软件上;它不会试图传播到其他系统。

### Description:

Non-replicating malicious code only resides on the target system or software that is attacked; it does not attempt to spread to other systems.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 509

提交日期 2008-01-30---> 修改日期 2012-10-30

* **Weakness Name:** *en: --> Replicating Malicious Code (Virus or Worm) zh: -->复制恶意代码（病毒或蠕虫）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

复制恶意代码（包括病毒和蠕虫）会在成功破坏目标系统或软件后尝试攻击其他系统。

### Description:

Replicating malicious code, including viruses and worms, will attempt to attack other systems once it has successfully compromised the target system or software.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 510

提交日期 I`m don`t know---> 修改日期 2017-05-03

* **Weakness Name:** *en: --> Trapdoor zh: -->暗门*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

陷门是一段隐藏的代码，它响应特殊输入，允许用户访问资源而无需通过正常的安全执行机制。

### Description:

A trapdoor is a hidden piece of code that responds to a special input, allowing its user access to resources without passing through the normal security enforcement mechanism.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 511

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Logic/Time Bomb zh: -->逻辑/时间炸弹*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含的代码旨在在特定时间过去或满足某个逻辑条件时破坏软件（或其环境）的合法操作。

### Description:

The software contains code that is designed to disrupt the legitimate operation of the software (or its environment) when a certain time passes, or when a certain logical condition is met.

### 详细描述:

当引爆定时炸弹或逻辑炸弹时，它可能会执行拒绝服务，例如崩溃系统，删除关键数据或降低系统响应时间。这枚炸弹可能放在复制或非复制的特洛伊木马中。

### Extended Description:

When the time bomb or logic bomb is detonated, it may perform a denial of service such as crashing the system, deleting critical data, or degrading system response time. This bomb might be placed within either a replicating or non-replicating Trojan horse.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 512

提交日期 I`m don`t know---> 修改日期 2012-10-30

* **Weakness Name:** *en: --> Spyware zh: -->间谍软件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件收集有关人类用户或用户活动的个人身份信息，但软件使用除自身之外的其他资源访问此信息，并且不需要用户明确批准或直接输入软件。

### Description:

The software collects personally identifiable information about a human user or the user's activities, but the software accesses this information using other resources besides itself, and it does not require that user's explicit approval or direct input into the software.

### 详细描述:

“间谍软件”是一个常用术语，有许多定义和解释。通常，它意味着收集信息或安装人类用户可能不允许的功能的软件，如果他们完全了解软件采取的操作。例如，用户可能希望税务软件收集社会安全号码并在提交纳税申报表时将其包括在内，但同一用户不希望游戏软件从该税务软件的数据中获取社会安全号码。

### Extended Description:

"Spyware" is a commonly used term with many definitions and interpretations. In general, it is meant to software that collects information or installs functionality that human users might not allow if they were fully aware of the actions being taken by the software. For example, a user might expect that tax software would collect a social security number and include it when filing a tax return, but that same user would not expect gaming software to obtain the social security number from that tax software's data.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 514

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Covert Channel zh: -->隐蔽通道*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

隐蔽通道是一种可用于以系统设计者不想要的方式传输信息的路径。

### Description:

A covert channel is a path that can be used to transfer information in a way not intended by the system's designers.

### 详细描述:

通常，系统没有授权传输，也不知道它的发生。

### Extended Description:

Typically the system has not given authorization for the transmission and has no knowledge of its occurrence.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

隐蔽通道可以被认为是一种紧急资源，这意味着它不是最初预期的资源，但它存在于应用程序的行为中。

Weakness ID: 515

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Covert Storage Channel zh: -->隐蔽存储通道*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

隐蔽存储信道通过一个程序的位设置和另一个程序读取这些位来传输信息。这种情况与普通操作的区别在于比特用于传送编码信息。

### Description:

A covert storage channel transfers information through the setting of bits by one program and the reading of those bits by another. What distinguishes this case from that of ordinary operation is that the bits are used to convey encoded information.

### 详细描述:

当带外数据存储在消息中以用于存储器重用时，发生隐藏存储信道。隐蔽信道通常被分类为存储或定时信道。示例包括使用旨在仅保存审计信息以传达用户密码的文件 - 使用文件的名称或者与其相关联的状态位，所有用户都可以读取该信息以指示文件的内容。隐写信息以这样的方式隐藏信息，即除了预期接收者之外，没有人知道消息的存在，这是隐蔽存储信道的一个很好的例子。

### Extended Description:

Covert storage channels occur when out-of-band data is stored in messages for the purpose of memory reuse. Covert channels are frequently classified as either storage or timing channels. Examples would include using a file intended to hold only audit information to convey user passwords--using the name of a file or perhaps status bits associated with it that can be read by all users to signal the contents of the file. Steganography, concealing information in such a manner that no one but the intended recipient knows of the existence of the message, is a good example of a covert storage channel.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 516

提交日期 I`m don`t know---> 修改日期 2009-10-29

* **Weakness Name:** *en: --> DEPRECATED (Duplicate): Covert Timing Channel zh: -->已弃用（重复）：隐蔽时间通道*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这个弱点可以在CWE-385找到。

### Description:

This weakness can be found at CWE-385.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 520

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> .NET Misconfiguration: Use of Impersonation zh: -->.NET配置错误：使用模拟*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

允许.NET应用程序在潜在升级的底层操作和文件系统访问级别运行可能很危险，并导致各种形式的攻击。

### Description:

Allowing a .NET application to run at potentially escalated levels of access to the underlying operating and file systems can be dangerous and result in various forms of attacks.

### 详细描述:

.NET服务器应用程序可以选择使用对客户端进行身份验证的用户的身份来执行。此功能的目的是绕过.NET应用程序代码中的身份验证和访问控制检查。身份验证由底层Web服务器（Microsoft Internet信息服务IIS）完成，该服务器将经过身份验证的令牌或未经身份验证的匿名令牌传递给.NET应用程序。使用令牌模拟客户端，应用程序然后依赖于NTFS目录和文件中的设置来控制访问。模拟使应用程序能够在运行.NET应用程序的服务器上执行代码并在经过身份验证和授权的用户的上下文中访问资源。

### Extended Description:

.NET server applications can optionally execute using the identity of the user authenticated to the client. The intention of this functionality is to bypass authentication and access control checks within the .NET application code. Authentication is done by the underlying web server (Microsoft Internet Information Service IIS), which passes the authenticated token, or unauthenticated anonymous token, to the .NET application. Using the token to impersonate the client, the application then relies on the settings within the NTFS directories and files to control access. Impersonation enables the application, on the server running the .NET application, to both execute code and access resources in the context of the authenticated and authorized user.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 521

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Weak Password Requirements zh: -->密码要求低*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品不要求用户具有强密码，这使攻击者更容易破坏用户帐户。

### Description:

The product does not require that users should have strong passwords, which makes it easier for attackers to compromise user accounts.

### 详细描述:

身份验证机制仅与其凭据一样强大。因此，要求用户拥有强密码非常重要。在尝试猜测用户密码时，缺乏密码复杂性会大大减少搜索空间，从而使暴力攻击更容易。

### Extended Description:

An authentication mechanism is only as strong as its credentials. For this reason, it is important to require users to have strong passwords. Lack of password complexity significantly reduces the search space when trying to guess user's passwords, making brute-force attacks easier.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 522

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Insufficiently Protected Credentials zh: -->受保护的凭据不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当应用程序传输或存储认证凭证并使用易受未经授权的拦截和/或检索的不安全方法时，就会出现这种弱点。

### Description:

This weakness occurs when the application transmits or stores authentication credentials and uses an insecure method that is susceptible to unauthorized interception and/or retrieval.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 523

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Unprotected Transport of Credentials zh: -->不受保护的证书传输*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

登录页面在从客户端到服务器的传输过程中没有使用足够的措施来保护用户名和密码。

### Description:

Login pages not using adequate measures to protect the user name and password while they are in transit from the client to the server.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

SSL（安全套接字层）为HTTP提供数据机密性和完整性。通过加密HTTP消息，SSL可以防止攻击者窃听或更改消息内容。

### 笔记 (Notes):

没有笔记

Weakness ID: 524

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Caching zh: -->通过缓存信息曝光*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序使用缓存来维护对象，线程，连接，页面或密码池，以最大限度地减少访问它们所需的时间或它们连接的资源。如果实施不当，这些缓存可能允许访问未经授权的信息或导致拒绝服务漏洞。

### Description:

The application uses a cache to maintain a pool of objects, threads, connections, pages, or passwords to minimize the time it takes to access them or the resources to which they connect. If implemented improperly, these caches can allow access to unauthorized information or cause a denial of service vulnerability.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 525

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Browser Caching zh: -->通过浏览器缓存进行信息曝光*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

对于每个网页，应用程序应具有适当的缓存策略，指定应缓存页面及其表单字段的程度。

### Description:

For each web page, the application should have an appropriate caching policy specifying the extent to which the page and its form fields should be cached.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 526

提交日期 2011-03-29---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Information Exposure Through Environmental Variables zh: -->通过环境变量进行信息曝光*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

环境变量可能包含有关远程服务器的敏感信息。

### Description:

Environmental variables may contain sensitive information about a remote server.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 527

提交日期 2009-12-28---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Exposure of CVS Repository to an Unauthorized Control Sphere zh: -->将CVS存储库暴露给未经授权的控制球体*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品将CVS存储库存储在目标或其他容器中，该目标或其他容器可供预期控制范围之外的actor访问。

### Description:

The product stores a CVS repository in a directory or other container that is accessible to actors outside of the intended control sphere.

### 详细描述:

攻击者可以恢复Web服务器或其他服务器上的CVS子目录中包含的信息，并将其用于恶意目的。此信息可能包括用户名，文件名，路径根和IP地址。

### Extended Description:

Information contained within a CVS subdirectory on a web server or other server could be recovered by an attacker and used for malicious purposes. This information may include usernames, filenames, path root, and IP addresses.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 528

提交日期 2009-12-28---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Exposure of Core Dump File to an Unauthorized Control Sphere zh: -->核心转储文件暴露给未经授权的控制领域*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品在目录中生成核心转储文件，该目录可供预期控制范围之外的actor访问。

### Description:

The product generates a core dump file in a directory that is accessible to actors outside of the intended control sphere.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 529

提交日期 2009-12-28---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Exposure of Access Control List Files to an Unauthorized Control Sphere zh: -->将访问控制列表文件暴露给未经授权的控制领域*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品将访问控制列表文件存储在目标或其他容器中，该容器可供预期控制范围之外的actor访问。

### Description:

The product stores access control list files in a directory or other container that is accessible to actors outside of the intended control sphere.

### 详细描述:

暴露这些访问控制列表文件可能会向攻击者提供有关站点或系统配置的信息。然后，该信息可用于绕过预期的安全策略或识别可从其发起攻击的可信系统。

### Extended Description:

Exposure of these access control list files may give the attacker information about the configuration of the site or system. This information may then be used to bypass the intended security policy or identify trusted systems from which an attack can be launched.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 530

提交日期 2009-12-28---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Exposure of Backup File to an Unauthorized Control Sphere zh: -->将备份文件暴露给未经授权的控制领域*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

备份文件存储在目标控制范围之外的actor可访问的目录中。

### Description:

A backup file is stored in a directory that is accessible to actors outside of the intended control sphere.

### 详细描述:

通常，使用.~bk等扩展名重命名旧文件，以区别于生产文件。通常可以检索以这种方式重命名并留在webroot中的旧文件的源代码。此重命名可能已由Web服务器自动执行，或由管理员手动执行。

### Extended Description:

Often, old files are renamed with an extension such as .~bk to distinguish them from production files. The source code for old files that have been renamed in this manner and left in the webroot can often be retrieved. This renaming may have been performed automatically by the web server, or manually by the administrator.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 531

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Test Code zh: -->通过测试代码的信息曝光*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

可访问的测试应用程序可能带来各种安全风险。由于开发人员或管理员很少考虑除了他们自己以外的人甚至会知道这些应用程序的存在，因此他们通常会包含敏感信息或功能。

### Description:

Accessible test applications can pose a variety of security risks. Since developers or administrators rarely consider that someone besides themselves would even know about the existence of these applications, it is common for them to contain sensitive information or functions.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 532

提交日期 2011-03-29---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Information Exposure Through Log Files zh: -->通过日志文件的信息曝光*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

写入日志文件的信息可能具有敏感性，可为攻击者提供有价值的指导或暴露敏感的用户信息。

### Description:

Information written to log files can be of a sensitive nature and give valuable guidance to an attacker or expose sensitive user information.

### 详细描述:

虽然在开发阶段记录所有信息可能会有所帮助，但在产品发布之前适当设置日志记录级别非常重要，这样敏感的用户数据和系统信息不会意外地暴露给潜在的攻击者。  
可以生成和存储不同的日志文件：  
  
服务器日志文件（例如server.log）。这可以提供有关文件的任何应用程序的信息。通常，这可以提供完整的路径名称和系统信息，有时也可以提供用户名和密码。  
用于调试的日志文件

### Extended Description:

While logging all information may be helpful during development stages, it is important that logging levels be set appropriately before a product ships so that sensitive user data and system information are not accidentally exposed to potential attackers.  
Different log files may be produced and stored for:  
  
Server log files (e.g. server.log). This can give information on whatever application left the file. Usually this can give full path names and system information, and sometimes usernames and passwords.  
log files that are used for debugging

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 533

提交日期 2011-03-29---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> DEPRECATED: Information Exposure Through Server Log Files zh: -->已弃用：通过服务器日志文件进行信息泄露*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此条目已被弃用，因为它的抽象级别太低。见CWE-532。

### Description:

This entry has been deprecated because its abstraction was too low-level. See CWE-532.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 534

提交日期 2011-03-29---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> DEPRECATED: Information Exposure Through Debug Log Files zh: -->已弃用：通过调试日志文件进行信息泄露*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此条目已被弃用，因为它的抽象级别太低。见CWE-532。

### Description:

This entry has been deprecated because its abstraction was too low-level. See CWE-532.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 535

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Shell Error Message zh: -->信息暴露通过Shell错误消息*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

命令shell错误消息指示Web应用程序代码中存在未处理的异常。在许多情况下，攻击者可以利用导致这些错误的条件来获取对系统的未授权访问。

### Description:

A command shell error message indicates that there exists an unhandled exception in the web application code. In many cases, an attacker can leverage the conditions that cause these errors in order to gain unauthorized access to the system.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 536

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Servlet Runtime Error Message zh: -->通过Servlet运行时错误消息的信息泄露*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

servlet错误消息表明Web应用程序代码中存在未处理的异常，并可能为攻击者提供有用的信息。

### Description:

A servlet error message indicates that there exists an unhandled exception in your web application code and may provide useful information to an attacker.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 537

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Java Runtime Error Message zh: -->通过Java运行时错误消息的信息暴露*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在许多情况下，攻击者可以利用导致未处理的异常错误的条件来获得对系统的未授权访问。

### Description:

In many cases, an attacker can leverage the conditions that cause unhandled exception errors in order to gain unauthorized access to the system.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 538

提交日期 2009-12-28---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> File and Directory Information Exposure zh: -->文件和目录信息曝光*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品将敏感信息存储在目标控制范围之外的actor可访问的文件或目录中。

### Description:

The product stores sensitive information in files or directories that are accessible to actors outside of the intended control sphere.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

根据使用情况，这可能是一个弱点或类别。需要进一步研究其所有儿童，并且可能需要澄清整个子树。目前的组织主要基于敏感信息的暴露，而不是主要的弱点。

Weakness ID: 539

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Persistent Cookies zh: -->通过持久性Cookie进行信息曝光*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

持久性cookie是存储在浏览器硬盘上的cookie。这可能会导致安全性和隐私问题，具体取决于cookie中存储的信息以及访问方式。

### Description:

Persistent cookies are cookies that are stored on the browser's hard drive. This can cause security and privacy issues depending on the information stored in the cookie and how it is accessed.

### 详细描述:

Cookie是由Web应用程序发送但本地存储在浏览器中的一小部分数据。这使应用程序可以使用cookie在页面之间传递信息并存储变量信息。 Web应用程序控制cookie中存储的信息及其使用方式。存储在cookie中的典型信息类型是会话标识符，个性化和自定义信息，在极少数情况下甚至是用于启用自动登录的用户名。有两种不同类型的cookie：会话cookie和持久性cookie。会话cookie只存储在浏览器的内存中，不存储在任何地方，但持久性cookie存储在浏览器的硬盘上。

### Extended Description:

Cookies are small bits of data that are sent by the web application but stored locally in the browser. This lets the application use the cookie to pass information between pages and store variable information. The web application controls what information is stored in a cookie and how it is used. Typical types of information stored in cookies are session Identifiers, personalization and customization information, and in rare cases even usernames to enable automated logins. There are two different types of cookies: session cookies and persistent cookies. Session cookies just live in the browser's memory, and are not stored anywhere, but persistent cookies are stored on the browser's hard drive.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 540

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Source Code zh: -->信息通过源代码曝光*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web服务器上的源代码通常包含敏感信息，用户通常无法访问。

### Description:

Source code on a web server often contains sensitive information and should generally not be accessible to users.

### 详细描述:

在某些情况下，从区域或服务器中删除源代码至关重要。例如，在系统上获取Perl源代码允许攻击者理解脚本的逻辑并提取极其有用的信息，例如代码错误或登录和密码。

### Extended Description:

There are situations where it is critical to remove source code from an area or server. For example, obtaining Perl source code on a system allows an attacker to understand the logic of the script and extract extremely useful information such as code bugs or logins and passwords.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 541

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Include Source Code zh: -->信息曝光包括源代码*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果可以访问包含文件源，则该文件可以包含用户名和密码，以及与应用程序和系统有关的敏感信息。

### Description:

If an include file source is accessible, the file can contain usernames and passwords, as well as sensitive information pertaining to the application and system.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 542

提交日期 2011-03-29---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> DEPRECATED: Information Exposure Through Cleanup Log Files zh: -->已弃用：通过清理日志文件进行信息泄露*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此条目已被弃用，因为它的抽象级别太低。见CWE-532。

### Description:

This entry has been deprecated because its abstraction was too low-level. See CWE-532.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 543

提交日期 2010-09-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Singleton Pattern Without Synchronization in a Multithreaded Context zh: -->在多线程上下文中使用没有同步的单例模式*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在多线程环境中创建资源时，该软件使用单例模式。

### Description:

The software uses the singleton pattern when creating a resource within a multithreaded environment.

### 详细描述:

使用单例模式可能不是线程安全的。

### Extended Description:

The use of a singleton pattern may not be thread-safe.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 544

提交日期 2009-03-10---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Missing Standardized Error Handling Mechanism zh: -->缺少标准化错误处理机制*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不使用标准化方法来处理整个代码中的错误，这可能会引入不一致的错误处理和由此产生的缺陷。

### Description:

The software does not use a standardized method for handling errors throughout the code, which might introduce inconsistent error handling and resultant weaknesses.

### 详细描述:

如果应用程序单独处理错误消息，则逐个处理，这可能会导致错误处理不一致。错误的原因可能会丢失。此外，有关错误原因的详细信息可能会无意中返回给用户。

### Extended Description:

If the application handles error messages individually, on a one-by-one basis, this is likely to result in inconsistent error handling. The causes of errors may be lost. Also, detailed information about the causes of an error may be unintentionally returned to the user.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 545

提交日期 2008-04-11---> 修改日期 2017-05-03

* **Weakness Name:** *en: --> DEPRECATED: Use of Dynamic Class Loading zh: -->已弃用：使用动态类加载*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这个弱点已被弃用，因为它与CWE-470部分重叠，它描述了合法的程序员行为，其他部分需要集成到其他条目中。

### Description:

This weakness has been deprecated because it partially overlaps CWE-470, it describes legitimate programmer behavior, and other portions will need to be integrated into other entries.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 546

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Suspicious Comment zh: -->可疑的评论*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码包含表明存在错误，功能不完整或缺陷的注释。

### Description:

The code contains comments that suggest the presence of bugs, incomplete functionality, or weaknesses.

### 详细描述:

代码中的许多可疑注释（例如BUG，HACK，FIXME，LATER，LATER2，TODO）表示缺少安全功能和检查。其他人指出程序员应该修复的代码问题，例如硬编码变量，错误处理，不使用存储过程和性能问题。

### Extended Description:

Many suspicious comments, such as BUG, HACK, FIXME, LATER, LATER2, TODO, in the code indicate missing security functionality and checking. Others indicate code problems that programmers should fix, such as hard-coded variables, error handling, not using stored procedures, and performance issues.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 547

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Hard-coded, Security-relevant Constants zh: -->使用硬编码，与安全相关的常量*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序使用硬编码常量而不是符号名称来表示安全关键值，这增加了代码维护或安全策略更改期间出错的可能性。

### Description:

The program uses hard-coded constants instead of symbolic names for security-critical values, which increases the likelihood of mistakes during code maintenance or security policy change.

### 详细描述:

如果开发人员未发现所有出现的硬编码常量，则如果其中一个常量未更改，则可能会做出错误的策略决策。更改这些值将需要在系统发布到现场后很难或不可能进行代码更改。此外，如果代码被公开，这些硬编码值可能会被攻击者使用。

### Extended Description:

If the developer does not find all occurrences of the hard-coded constants, an incorrect policy decision may be made if one of the constants is not changed. Making changes to these values will require code changes that may be difficult or impossible once the system is released to the field. In addition, these hard-coded values may become available to attackers if the code is ever disclosed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 548

提交日期 2011-03-29---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Information Exposure Through Directory Listing zh: -->通过目录列表进行信息曝光*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

目录列表被不恰当地暴露，从而向攻击者提供潜在的敏感信息。

### Description:

A directory listing is inappropriately exposed, yielding potentially sensitive information to attackers.

### 详细描述:

目录列表为攻击者提供了位于目录内的所有资源的完整索引。具体的风险和后果取决于列出和访问的文件。

### Extended Description:

A directory listing provides an attacker with the complete index of all the resources located inside of the directory. The specific risks and consequences vary depending on which files are listed and accessible.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 549

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Missing Password Field Masking zh: -->缺少密码字段屏蔽*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在进入期间不会屏蔽密码，从而增加了攻击者观察和捕获密码的可能性。

### Description:

The software does not mask passwords during entry, increasing the potential for attackers to observe and capture passwords.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 550

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Server Error Message zh: -->通过服务器错误消息暴露信息*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

某些条件（如网络故障）将导致显示服务器错误消息。

### Description:

Certain conditions, such as network failure, will cause a server error message to be displayed.

### 详细描述:

虽然错误信息本身并不危险，但攻击者可以从中收集错误信息，这可能会导致最终的问题。

### Extended Description:

While error messages in and of themselves are not dangerous, per se, it is what an attacker can glean from them that might cause eventual problems.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 551

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incorrect Behavior Order: Authorization Before Parsing and Canonicalization zh: -->行为顺序不正确：解析和规范化之前的授权*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果Web服务器在检查请求的URL之前没有完全解析它们以进行授权，则攻击者可能会绕过授权保护。

### Description:

If a web server does not fully parse requested URLs before it examines them for authorization, it may be possible for an attacker to bypass authorization protection.

### 详细描述:

例如，字符串/./和/都表示当前目录。如果/ SomeDirectory是受保护的目录而攻击者请求/./SomeDirectory，则如果/./未在执行授权检查之前转换为/，则攻击者可能能够访问该资源。

### Extended Description:

For instance, the character strings /./ and / both mean current directory. If /SomeDirectory is a protected directory and an attacker requests /./SomeDirectory, the attacker may be able to gain access to the resource if /./ is not converted to / before the authorization check is performed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 552

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Files or Directories Accessible to External Parties zh: -->可供外部各方访问的文件或目录*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

文件或目录可以在不应该的环境中访问。

### Description:

Files or directories are accessible in the environment that should not be.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 553

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Command Shell in Externally Accessible Directory zh: -->外部可访问目录中的命令外壳*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

/ cgi-bin /或其他可访问目录中存在可能的shell文件。这非常危险，攻击者可以使用它来执行Web服务器上的命令。

### Description:

A possible shell file exists in /cgi-bin/ or other accessible directories. This is extremely dangerous and can be used by an attacker to execute commands on the web server.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 554

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> ASP.NET Misconfiguration: Not Using Input Validation Framework zh: -->ASP.NET配置错误：不使用输入验证框架*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

ASP.NET应用程序不使用输入验证框架。

### Description:

The ASP.NET application does not use an input validation framework.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 555

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> J2EE Misconfiguration: Plaintext Password in Configuration File zh: -->J2EE配置错误：配置文件中的纯文本密码*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

J2EE应用程序将明文密码存储在配置文件中。

### Description:

The J2EE application stores a plaintext password in a configuration file.

### 详细描述:

在配置文件中存储明文密码允许任何能够读取该文件的人访问受密码保护的资源，使其成为攻击者的轻松目标。

### Extended Description:

Storing a plaintext password in a configuration file allows anyone who can read the file to access the password-protected resource, making it an easy target for attackers.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 556

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> ASP.NET Misconfiguration: Use of Identity Impersonation zh: -->ASP.NET配置错误：使用身份模拟*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

配置ASP.NET应用程序以使用模拟凭据运行可能会为应用程序提供不必要的权限。

### Description:

Configuring an ASP.NET application to run with impersonated credentials may give the application unnecessary privileges.

### 详细描述:

使用模拟凭据允许ASP.NET应用程序以其正在执行的客户端的权限或在其配置中授予的任意权限运行。

### Extended Description:

The use of impersonated credentials allows an ASP.NET application to run with either the privileges of the client on whose behalf it is executing or with arbitrary privileges granted in its configuration.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 558

提交日期 2008-04-11---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Use of getlogin() in Multithreaded Application zh: -->在多线程应用程序中使用getlogin（）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序在多线程上下文中使用getlogin（）函数，可能导致它返回不正确的值。

### Description:

The application uses the getlogin() function in a multithreaded context, potentially causing it to return incorrect values.

### 详细描述:

getlogin（）函数返回一个指向字符串的指针，该字符串包含与调用进程关联的用户的名称。该函数不是可重入的，这意味着如果从另一个进程调用它，则内容不会被锁定，并且字符串的值可以被另一个进程更改。这使得使用起来非常危险，因为其他进程可以更改用户名，因此无法信任该函数的结果。

### Extended Description:

The getlogin() function returns a pointer to a string that contains the name of the user associated with the calling process. The function is not reentrant, meaning that if it is called from another process, the contents are not locked out and the value of the string can be changed by another process. This makes it very risky to use because the username can be changed by other processes, so the results of the function cannot be trusted.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 560

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of umask() with chmod-style Argument zh: -->使用带有chmod样式参数的umask（）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用不正确的参数调用umask（），该参数被指定为chmod（）的参数。

### Description:

The product calls umask() with an incorrect argument that is specified as if it is an argument to chmod().

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

umask（）手册页以false语句开头：“umask将umask设置为mask＆0777”虽然这种行为最好与chmod（）的用法一致，其中用户提供的参数指定要在指定文件上启用的位，umask（）的行为实际上是相反的：umask（）将umask设置为~mask＆0777。umask（）手册页继续描述umask（）的正确用法：“umask由open使用（ ）在新创建的文件上设置初始文件权限。具体来说，umask中的权限从mode参数关闭为open（2）（因此，例如，公共umask默认值022导致创建新文件）权限0666＆~022 = 0644 = rw-r - r--在通常情况下，模式被指定为0666）。“

Weakness ID: 561

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Dead Code zh: -->死代码*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含死代码，永远不能执行。

### Description:

The software contains dead code, which can never be executed.

### 详细描述:

死代码是永远不能在正在运行的程序中执行的源代码。周围的代码使得一段代码无法执行。

### Extended Description:

Dead code is source code that can never be executed in a running program. The surrounding code makes it impossible for a section of code to ever be executed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 562

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Return of Stack Variable Address zh: -->堆栈变量地址的返回*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

函数返回堆栈变量的地址，这将导致意外的程序行为，通常以崩溃的形式。

### Description:

A function returns the address of a stack variable, which will cause unintended program behavior, typically in the form of a crash.

### 详细描述:

因为局部变量是在堆栈上分配的，所以当程序返回指向局部变量的指针时，它返回一个堆栈地址。后续函数调用可能会重复使用相同的堆栈地址，从而覆盖指针的值，该值不再对应于同一个变量，因为函数的堆栈帧在返回时无效。充其量这将导致指针的值意外更改。在许多情况下，它会导致程序在下次取消引用指针时崩溃。

### Extended Description:

Because local variables are allocated on the stack, when a program returns a pointer to a local variable, it is returning a stack address. A subsequent function call is likely to re-use this same stack address, thereby overwriting the value of the pointer, which no longer corresponds to the same variable since a function's stack frame is invalidated when it returns. At best this will cause the value of the pointer to change unexpectedly. In many cases it causes the program to crash the next time the pointer is dereferenced.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 563

提交日期 2014-06-23---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Assignment to Variable without Use zh: -->不使用而赋值给变量*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

变量的值已分配但从未使用过，使其成为死存储。

### Description:

The variable's value is assigned but never used, making it a dead store.

### 详细描述:

赋值后，变量要么被赋予另一个值，要么超出范围。变量很可能只是残留，但未使用的变量也可能指出错误。

### Extended Description:

After the assignment, the variable is either assigned another value or goes out of scope. It is likely that the variable is simply vestigial, but it is also possible that the unused variable points out a bug.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 564

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> SQL Injection: Hibernate zh: -->SQL注入：Hibernate*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

使用Hibernate执行使用用户控制的输入构建的动态SQL语句可以允许攻击者修改语句的含义或执行任意SQL命令。

### Description:

Using Hibernate to execute a dynamic SQL statement built with user-controlled input can allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 565

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Reliance on Cookies without Validation and Integrity Checking zh: -->依赖于Cookie而无需验证和完整性检查*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序在执行安全关键操作时依赖于cookie的存在或值，但它无法正确确保该设置对关联用户有效。

### Description:

The application relies on the existence or values of cookies when performing security-critical operations, but it does not properly ensure that the setting is valid for the associated user.

### 详细描述:

攻击者可以在浏览器中轻松修改cookie，也可以在浏览器外部实现客户端代码。在没有详细验证和完整性检查的情况下依赖cookie可以允许攻击者绕过身份验证，执行注入攻击（如SQL注入和跨站点脚本），或以其他方式修改输入。

### Extended Description:

Attackers can easily modify cookies, within the browser or by implementing the client-side code outside of the browser. Reliance on cookies without detailed validation and integrity checking can allow attackers to bypass authentication, conduct injection attacks such as SQL injection and cross-site scripting, or otherwise modify inputs in unexpected ways.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此问题可能是Web应用程序中许多类型的弱点的主要问题。开发人员可以对URL参数执行适当的验证，同时假设攻击者无法修改cookie。因此，程序可能会跳过基本输入验证，以启用跨站点脚本，SQL注入，价格篡改和其他攻击。

Weakness ID: 566

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Authorization Bypass Through User-Controlled SQL Primary Key zh: -->授权绕过用户控制的SQL主键*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用一个数据库表，该表包含一个演员不应该访问的记录，但它执行一个带有可由该actor控制的主键的SQL语句。

### Description:

The software uses a database table that includes records that should not be accessible to an actor, but it executes a SQL statement with a primary key that can be controlled by that actor.

### 详细描述:

当用户可以将主键设置为任何值时，用户可以修改该键以指向未授权的记录。  
数据库访问控制错误发生在：  
  
  
数据从不受信任的来源进入程序。  
该数据用于指定SQL查询中主键的值。  
不受信任的源没有权限可以访问关联表中的所有行。

### Extended Description:

When a user can set a primary key to any value, then the user can modify the key to point to unauthorized records.  
Database access control errors occur when:  
  
  
Data enters a program from an untrusted source.  
The data is used to specify the value of a primary key in a SQL query.  
The untrusted source does not have the permissions to be able to access all rows in the associated table.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 567

提交日期 2010-12-13---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Unsynchronized Access to Shared Data in a Multithreaded Context zh: -->多线程上下文中对共享数据的不同步访问*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品无法正确同步共享数据，例如跨线程的静态变量，这可能导致未定义的行为和不可预测的数据更改。

### Description:

The product does not properly synchronize shared data, such as static variables across threads, which can lead to undefined behavior and unpredictable data changes.

### 详细描述:

在servlet中，共享静态变量不受并发访问的保护，但servlet是多线程的。这是J2EE应用程序中的典型编程错误，因为多线程由框架处理。当共享变量可能受到攻击者的影响时，一个线程最终可能会修改变量以包含对同样使用变量中的数据的其他线程无效的数据。  
请注意，这个弱点并不是servlet独有的。

### Extended Description:

Within servlets, shared static variables are not protected from concurrent access, but servlets are multithreaded. This is a typical programming mistake in J2EE applications, since the multithreading is handled by the framework. When a shared variable can be influenced by an attacker, one thread could wind up modifying the variable to contain data that is not valid for a different thread that is also using the data within the variable.  
Note that this weakness is not unique to servlets.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 568

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> finalize() Method Without super.finalize() zh: -->finalize（）方法没有super.finalize（）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个不调用super.finalize（）的finalize（）方法。

### Description:

The software contains a finalize() method that does not call super.finalize().

### 详细描述:

Java语言规范声明，finalize（）方法调用super.finalize（）是一种很好的做法。

### Extended Description:

The Java Language Specification states that it is a good practice for a finalize() method to call super.finalize().

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 570

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Expression is Always False zh: -->表达始终是错误的*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个总是评估为false的表达式。

### Description:

The software contains an expression that will always evaluate to false.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 571

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Expression is Always True zh: -->表达始终如一*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个总是评估为true的表达式。

### Description:

The software contains an expression that will always evaluate to true.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 572

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Call to Thread run() instead of start() zh: -->调用Thread run（）而不是start（）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

程序调用线程的run（）方法而不是调用start（），这会导致代码在调用者的线程而不是被调用者中运行。

### Description:

The program calls a thread's run() method instead of calling start(), which causes the code to run in the thread of the caller instead of the callee.

### 详细描述:

在大多数情况下，直接调用Thread对象的run（）方法是一个错误。程序员打算开始一个新的控制线程，但是不小心调用了run（）而不是start（），因此run（）方法将在调用者的控制线程中执行。

### Extended Description:

In most cases a direct call to a Thread object's run() method is a bug. The programmer intended to begin a new thread of control, but accidentally called run() instead of start(), so the run() method will execute in the caller's thread of control.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 573

提交日期 2011-03-29---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Following of Specification by Caller zh: -->调用者对规范的不正确遵循*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不遵循或错误地遵循实现语言，环境，框架，协议或平台所要求的规范。

### Description:

The software does not follow or incorrectly follows the specifications as required by the implementation language, environment, framework, protocol, or platform.

### 详细描述:

当利用外部功能（例如API）时，调用者根据外部功能的要求这样做是很重要的，否则可能会导致意外行为，从而可能使系统容易受到任何数量的攻击。

### Extended Description:

When leveraging external functionality, such as an API, it is important that the caller does so in accordance with the requirements of the external functionality or else unintended behaviors may result, possibly leaving the system vulnerable to any number of exploits.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 574

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> EJB Bad Practices: Use of Synchronization Primitives zh: -->EJB不良做法：使用同步原语*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序通过使用线程同步原语违反Enterprise JavaBeans（EJB）规范。

### Description:

The program violates the Enterprise JavaBeans (EJB) specification by using thread synchronization primitives.

### 详细描述:

Enterprise JavaBeans规范要求每个bean提供程序遵循一组编程指南，这些指南旨在确保bean在任何EJB容器中都是可移植的并且行为一致。在这种情况下，该程序违反了以下EJB准则：“企业bean不得使用线程同步原语来同步多个实例的执行。”规范以下列方式证明了这一要求：“这条规则是确保一致的运行时语义所必需的，因为虽然一些EJB容器可能使用单个JVM来执行所有企业bean的实例，但其他EJB容器可能会将实例分布在多个JVM上。”

### Extended Description:

The Enterprise JavaBeans specification requires that every bean provider follow a set of programming guidelines designed to ensure that the bean will be portable and behave consistently in any EJB container. In this case, the program violates the following EJB guideline: "An enterprise bean must not use thread synchronization primitives to synchronize execution of multiple instances." The specification justifies this requirement in the following way: "This rule is required to ensure consistent runtime semantics because while some EJB containers may use a single JVM to execute all enterprise bean's instances, others may distribute the instances across multiple JVMs."

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 575

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> EJB Bad Practices: Use of AWT Swing zh: -->EJB不良做法：使用AWT Swing*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序使用AWT / Swing违反了Enterprise JavaBeans（EJB）规范。

### Description:

The program violates the Enterprise JavaBeans (EJB) specification by using AWT/Swing.

### 详细描述:

Enterprise JavaBeans规范要求每个bean提供程序遵循一组编程指南，这些指南旨在确保bean在任何EJB容器中都是可移植的并且行为一致。在这种情况下，程序违反了以下EJB准则：“企业bean不得使用AWT功能尝试将信息输出到显示器，或从键盘输入信息。”该规范通过以下方式证明了这一要求：“大多数服务器不允许应用程序与连接到服务器系统的键盘/显示器之间的直接交互。”

### Extended Description:

The Enterprise JavaBeans specification requires that every bean provider follow a set of programming guidelines designed to ensure that the bean will be portable and behave consistently in any EJB container. In this case, the program violates the following EJB guideline: "An enterprise bean must not use the AWT functionality to attempt to output information to a display, or to input information from a keyboard." The specification justifies this requirement in the following way: "Most servers do not allow direct interaction between an application program and a keyboard/display attached to the server system."

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 576

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> EJB Bad Practices: Use of Java I/O zh: -->EJB不良做法：使用Java I / O.*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序使用java.io包违反Enterprise JavaBeans（EJB）规范。

### Description:

The program violates the Enterprise JavaBeans (EJB) specification by using the java.io package.

### 详细描述:

Enterprise JavaBeans规范要求每个bean提供程序遵循一组编程指南，这些指南旨在确保bean在任何EJB容器中都是可移植的并且行为一致。在这种情况下，该程序违反了以下EJB准则：“企业bean不得使用java.io包来尝试访问文件系统中的文件和目录。”该规范以下列方式证明了这一要求：“文件系统API不适合业务组件访问数据。业务组件应使用资源管理器API（如JDBC）来存储数据。”

### Extended Description:

The Enterprise JavaBeans specification requires that every bean provider follow a set of programming guidelines designed to ensure that the bean will be portable and behave consistently in any EJB container. In this case, the program violates the following EJB guideline: "An enterprise bean must not use the java.io package to attempt to access files and directories in the file system." The specification justifies this requirement in the following way: "The file system APIs are not well-suited for business components to access data. Business components should use a resource manager API, such as JDBC, to store data."

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 577

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> EJB Bad Practices: Use of Sockets zh: -->EJB不良做法：使用套接字*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序使用套接字违反Enterprise JavaBeans（EJB）规范。

### Description:

The program violates the Enterprise JavaBeans (EJB) specification by using sockets.

### 详细描述:

Enterprise JavaBeans规范要求每个bean提供程序遵循一组编程指南，这些指南旨在确保bean在任何EJB容器中都是可移植的并且行为一致。在这种情况下，该程序违反了以下EJB准则：“企业bean不得尝试侦听套接字，接受套接字上的连接，或使用套接字进行多播。”规范通过以下方式证明了这一要求：“EJB体系结构允许企业bean实例成为网络套接字客户端，但它不允许它成为网络服务器。允许实例成为网络服务器会与企业bean的基本功能 - 为EJB客户端提供服务。“

### Extended Description:

The Enterprise JavaBeans specification requires that every bean provider follow a set of programming guidelines designed to ensure that the bean will be portable and behave consistently in any EJB container. In this case, the program violates the following EJB guideline: "An enterprise bean must not attempt to listen on a socket, accept connections on a socket, or use a socket for multicast." The specification justifies this requirement in the following way: "The EJB architecture allows an enterprise bean instance to be a network socket client, but it does not allow it to be a network server. Allowing the instance to become a network server would conflict with the basic function of the enterprise bean-- to serve the EJB clients."

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 578

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> EJB Bad Practices: Use of Class Loader zh: -->EJB不良做法：使用类加载器*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序通过使用类加载器违反Enterprise JavaBeans（EJB）规范。

### Description:

The program violates the Enterprise JavaBeans (EJB) specification by using the class loader.

### 详细描述:

Enterprise JavaBeans规范要求每个bean提供程序遵循一组编程指南，这些指南旨在确保bean在任何EJB容器中都是可移植的并且行为一致。在这种情况下，程序违反了以下EJB准则：“企业bean不得尝试创建类加载器;获取当前类加载器;设置上下文类加载器;设置安全管理器;创建新的安全管理器;停止JVM ;或更改输入，输出和错误流。“规范通过以下方式证明了这一要求：“这些函数是为EJB容器保留的。允许企业bean使用这些函数可能会危及安全性并降低容器正确管理运行时环境的能力。”

### Extended Description:

The Enterprise JavaBeans specification requires that every bean provider follow a set of programming guidelines designed to ensure that the bean will be portable and behave consistently in any EJB container. In this case, the program violates the following EJB guideline: "The enterprise bean must not attempt to create a class loader; obtain the current class loader; set the context class loader; set security manager; create a new security manager; stop the JVM; or change the input, output, and error streams." The specification justifies this requirement in the following way: "These functions are reserved for the EJB container. Allowing the enterprise bean to use these functions could compromise security and decrease the container's ability to properly manage the runtime environment."

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 579

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> J2EE Bad Practices: Non-serializable Object Stored in Session zh: -->J2EE不良做法：会话中存储的非序列化对象*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序将非序列化对象存储为HttpSession属性，这会损害可靠性。

### Description:

The application stores a non-serializable object as an HttpSession attribute, which can hurt reliability.

### 详细描述:

J2EE应用程序可以使用多个JVM，以提高应用程序的可靠性和性能。为了使多个JVM作为单个应用程序显示给最终用户，J2EE容器可以跨多个JVM复制HttpSession对象，这样，如果一个JVM变得不可用，另一个JVM就可以介入并取代它而不会中断应用程序的流程。这只有在所有会话数据都可序列化的情况下才有可能，允许在JVM之间复制会话。

### Extended Description:

A J2EE application can make use of multiple JVMs in order to improve application reliability and performance. In order to make the multiple JVMs appear as a single application to the end user, the J2EE container can replicate an HttpSession object across multiple JVMs so that if one JVM becomes unavailable another can step in and take its place without disrupting the flow of the application. This is only possible if all session data is serializable, allowing the session to be duplicated between the JVMs.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 580

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> clone() Method Without super.clone() zh: -->clone（）方法没有super.clone（）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个clone（）方法，该方法不会调用super.clone（）来获取新对象。

### Description:

The software contains a clone() method that does not call super.clone() to obtain the new object.

### 详细描述:

clone（）的所有实现都应该通过调用super.clone（）来获取新对象。如果某个类不遵循此约定，则子类的clone（）方法将返回错误类型的对象。

### Extended Description:

All implementations of clone() should obtain the new object by calling super.clone(). If a class does not follow this convention, a subclass's clone() method will return an object of the wrong type.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 581

提交日期 2008-01-30---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Object Model Violation: Just One of Equals and Hashcode Defined zh: -->对象模型违规：只定义一个等于和Hashcode*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不保持相等对象的相等哈希码。

### Description:

The software does not maintain equal hashcodes for equal objects.

### 详细描述:

期望Java对象遵守与平等相关的许多不变量。其中一个不变量是等对象必须具有相同的哈希码。换句话说，如果a.equals（b）== true，则a.hashCode（）== b.hashCode（）。

### Extended Description:

Java objects are expected to obey a number of invariants related to equality. One of these invariants is that equal objects must have equal hashcodes. In other words, if a.equals(b) == true then a.hashCode() == b.hashCode().

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 582

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Array Declared Public, Final, and Static zh: -->Array声明为Public，Final和Static*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

程序声明一个数组public，final和static，这不足以防止修改数组的内容。

### Description:

The program declares an array public, final, and static, which is not sufficient to prevent the array's contents from being modified.

### 详细描述:

因为数组是可变对象，所以最终约束要求数组对象本身只分配一次，但不保证数组元素的值。由于阵列是公共的，因此恶意程序可以更改存储在阵列中的值。因此，在大多数情况下，声明为public，final和static的数组是一个错误。

### Extended Description:

Because arrays are mutable objects, the final constraint requires that the array object itself be assigned only once, but makes no guarantees about the values of the array elements. Since the array is public, a malicious program can change the values stored in the array. As such, in most cases an array declared public, final and static is a bug.

### 问题背景 (Background Detail):

移动代码（在本例中为Java Applet）是通过网络传输并在远程计算机上执行的代码。由于移动代码开发人员几乎无法控制其代码执行的环境，因此特殊的安全问题变得相关。最大的环境威胁之一是移动代码与其他可能是恶意的移动代码并行运行的风险。因为所有流行的Web浏览器都在同一个JVM中一起执行来自多个源的代码，所以移动代码的许多安全准则都集中在防止有权访问同一虚拟机的对手操纵对象的状态和行为。程序正在运行。

### 笔记 (Notes):

没有笔记

Weakness ID: 583

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> finalize() Method Declared Public zh: -->finalize（）方法声明为Public*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序通过声明finalize（）方法公开来违反移动代码的安全编码原则。

### Description:

The program violates secure coding principles for mobile code by declaring a finalize() method public.

### 详细描述:

除了在finalize（）的实现中调用super.finalize（）之外，程序绝不应该显式调用finalize。在移动代码情况下，如果攻击者可以恶意调用您的一个finalize（）方法，那么手动垃圾收集的其他容易出错的做法可能会成为安全威胁，因为它是通过公共访问声明的。

### Extended Description:

A program should never call finalize explicitly, except to call super.finalize() inside an implementation of finalize(). In mobile code situations, the otherwise error prone practice of manual garbage collection can become a security threat if an attacker can maliciously invoke one of your finalize() methods because it is declared with public access.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 584

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Return Inside Finally Block zh: -->返回内部最后阻止*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码在finally块中有一个return语句，这将导致try块中的任何抛出异常被丢弃。

### Description:

The code has a return statement inside a finally block, which will cause any thrown exception in the try block to be discarded.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 585

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Empty Synchronized Block zh: -->空同步块*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个空的同步块。

### Description:

The software contains an empty synchronized block.

### 详细描述:

空的同步块实际上不会实现任何同步，并且可能指示代码的问题部分。可以发生空的同步块，因为在不移除同步块的情况下，注释掉同步块内不再需要的代码。

### Extended Description:

An empty synchronized block does not actually accomplish any synchronization and may indicate a troubled section of code. An empty synchronized block can occur because code no longer needed within the synchronized block is commented out without removing the synchronized block.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 586

提交日期 2008-09-09---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Explicit Call to Finalize() zh: -->明确调用Finalize（）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件从终结器外部显式调用finalize（）方法。

### Description:

The software makes an explicit call to the finalize() method from outside the finalizer.

### 详细描述:

虽然Java语言规范允许从终结器外部调用对象的finalize（）方法，但这样做通常是个坏主意。例如，调用finalize（）显式意味着finalize（）将被多次调用：第一次是显式调用，最后一次是在对象被垃圾回收后进行的调用。

### Extended Description:

While the Java Language Specification allows an object's finalize() method to be called from outside the finalizer, doing so is usually a bad idea. For example, calling finalize() explicitly means that finalize() will be called more than once: the first time will be the explicit call and the last time will be the call that is made after the object is garbage collected.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 587

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Assignment of a Fixed Address to a Pointer zh: -->将固定地址分配给指针*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件设置指向除NULL或0以外的特定地址的指针。

### Description:

The software sets a pointer to a specific address other than NULL or 0.

### 详细描述:

使用固定地址是不可移植的，因为该地址可能在所有环境或平台中都无效。

### Extended Description:

Using a fixed address is not portable because that address will probably not be valid in all environments or platforms.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 588

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Attempt to Access Child of a Non-structure Pointer zh: -->尝试访问非结构指针的子项*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

将非结构类型转换为结构类型并访问字段可能导致内存访问错误或数据损坏。

### Description:

Casting a non-structure type to a structure type and accessing a field can lead to memory access errors or data corruption.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 589

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Call to Non-ubiquitous API zh: -->调用非普遍存在的API*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用目标平台的所有版本上不存在的API函数。这可能会导致可移植性问题或不一致，从而导致拒绝服务或其他后果。

### Description:

The software uses an API function that does not exist on all versions of the target platform. This could cause portability problems or inconsistencies that allow denial of service or other consequences.

### 详细描述:

某些提供OS支持的安全功能的功能在常用的所有OS版本上都不可用。同样，出于安全原因，函数经常被弃用或过时，不应使用。

### Extended Description:

Some functions that offer security features supported by the OS are not available on all versions of the OS in common use. Likewise, functions are often deprecated or made obsolete for security reasons and should not be used.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 590

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Free of Memory not on the Heap zh: -->没有记忆而不是堆*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序在指向未使用相关堆分配函数（如malloc（），calloc（）或realloc（）分配的内存的指针上调用free（）。

### Description:

The application calls free() on a pointer to memory that was not allocated using associated heap allocation functions such as malloc(), calloc(), or realloc().

### 详细描述:

当在无效指针上调用free（）时，程序的内存管理数据结构可能会损坏。这种损坏可能导致程序崩溃，或者在某些情况下，攻击者可能会导致free（）操作可控内存位置以修改关键程序变量或执行代码。

### Extended Description:

When free() is called on an invalid pointer, the program's memory management data structures may become corrupted. This corruption can cause the program to crash or, in some circumstances, an attacker may be able to cause free() to operate on controllable memory locations to modify critical program variables or execute code.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在C ++中，如果使用new运算符来分配内存，则可以在实现中使用malloc（），calloc（）或realloc（）系列函数进行分配。知道此行为的人可能会选择将此问题映射到CWE-590或其父级CWE-762，具体取决于他们的观点。

Weakness ID: 591

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Sensitive Data Storage in Improperly Locked Memory zh: -->锁定内存不正确的敏感数据存储*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序将敏感数据存储在未锁定或未正确锁定的内存中，这可能导致内存被虚拟内存管理器写入磁盘上的交换文件。这可以使外部参与者更容易访问数据。

### Description:

The application stores sensitive data in memory that is not locked, or that has been incorrectly locked, which might cause the memory to be written to swap files on disk by the virtual memory manager. This can make the data more accessible to external actors.

### 详细描述:

在Windows系统上，VirtualLock功能可以锁定一页内存，以确保它将保留在内存中而不会交换到磁盘。但是，在旧版本的Windows上，例如95,98或Me，VirtualLock（）函数只是一个存根，不提供保护。在POSIX系统上，mlock（）调用确保页面将驻留在内存中，但不保证页面不会出现在交换中。因此，它不适合用作敏感数据的保护机制。某些平台，特别是Linux，确实保证页面不会被交换，但这是非标准的并且不可移植。调用mlock（）也需要supervisor权限。必须检查这两个调用的返回值，以确保锁定操作实际上是成功的。

### Extended Description:

On Windows systems the VirtualLock function can lock a page of memory to ensure that it will remain present in memory and not be swapped to disk. However, on older versions of Windows, such as 95, 98, or Me, the VirtualLock() function is only a stub and provides no protection. On POSIX systems the mlock() call ensures that a page will stay resident in memory but does not guarantee that the page will not appear in the swap. Therefore, it is unsuitable for use as a protection mechanism for sensitive data. Some platforms, in particular Linux, do make the guarantee that the page will not be swapped, but this is non-standard and is not portable. Calls to mlock() also require supervisor privilege. Return values for both of these calls must be checked to ensure that the lock operation was actually successful.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 592

提交日期 2017-05-03---> 修改日期 2017-05-03

* **Weakness Name:** *en: --> DEPRECATED: Authentication Bypass Issues zh: -->DEPRECATED：身份验证绕过问题*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这个弱点已被弃用，因为它涵盖了CWE-287中已经描述的冗余概念。

### Description:

This weakness has been deprecated because it covered redundant concepts already described in CWE-287.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 593

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Authentication Bypass: OpenSSL CTX Object Modified after SSL Objects are Created zh: -->身份验证绕过：创建SSL对象后修改OpenSSL CTX对象*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

连接创建开始后，软件会修改SSL上下文。

### Description:

The software modifies the SSL context after connection creation has begun.

### 详细描述:

如果程序在从其创建SSL对象后修改SSL\_CTX对象，则从原始上下文创建的旧SSL对象可能都会受到该更改的影响。

### Extended Description:

If the program modifies the SSL\_CTX object after creating SSL objects from it, there is the possibility that older SSL objects created from the original context could all be affected by that change.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 594

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> J2EE Framework: Saving Unserializable Objects to Disk zh: -->J2EE框架：将不可序列化的对象保存到磁盘*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当J2EE容器尝试将不可序列化的对象写入磁盘时，无法保证该过程将成功完成。

### Description:

When the J2EE container attempts to write unserializable objects to disk there is no guarantee that the process will complete successfully.

### 详细描述:

在高负载条件下，大多数J2EE应用程序框架将对象刷新到磁盘以管理传入请求的内存要求。例如，会话范围对象，甚至是应用程序范围对象，在需要时写入磁盘。虽然这些应用程序框架完成了将对象写入磁盘的实际工作，但它们并未强制要求这些对象可序列化，从而使Web应用程序容易受到序列化失败引起的崩溃的影响。攻击者可以通过向服务器发送足够的请求来强制Web应用程序将对象保存到磁盘来安装拒绝服务攻击。

### Extended Description:

In heavy load conditions, most J2EE application frameworks flush objects to disk to manage memory requirements of incoming requests. For example, session scoped objects, and even application scoped objects, are written to disk when required. While these application frameworks do the real work of writing objects to disk, they do not enforce that those objects be serializable, thus leaving the web application vulnerable to crashes induced by serialization failure. An attacker may be able to mount a denial of service attack by sending enough requests to the server to force the web application to save objects to disk.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 595

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Comparison of Object References Instead of Object Contents zh: -->对象引用的比较而不是对象内容*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序比较对象引用而不是对象本身的内容，从而阻止它检测等效对象。

### Description:

The program compares object references instead of the contents of the objects themselves, preventing it from detecting equivalent objects.

### 详细描述:

例如，在Java中，使用==比较对象通常会产生欺骗性结果，因为==运算符会比较对象引用而不是值;通常，这意味着对字符串使用==实际上是比较字符串的引用，而不是它们的值。

### Extended Description:

For example, in Java, comparing objects using == usually produces deceptive results, since the == operator compares object references rather than values; often, this means that using == for strings is actually comparing the strings' references, not their values.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 596

提交日期 2008-04-11---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> DEPRECATED: Incorrect Semantic Object Comparison zh: -->弃用：语义对象比较不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

这个弱点已被弃用。它描述得很差，很难与其他条目区分开来。仅因特定领域的考虑而分配单独的ID也是不合适的。它最接近的等价物是CWE-1023。

### Description:

This weakness has been deprecated. It was poorly described and difficult to distinguish from other entries. It was also inappropriate to assign a separate ID solely because of domain-specific considerations. Its closest equivalent is CWE-1023.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 597

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Wrong Operator in String Comparison zh: -->在字符串比较中使用错误的运算符*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

比较字符串时，产品使用错误的运算符，例如，当使用equals（）方法时，使用“==”。

### Description:

The product uses the wrong operator when comparing a string, such as using "==" when the equals() method should be used instead.

### 详细描述:

在Java中，使用==或！=比较两个字符串的相等性实际上比较了两个对象的相等性，而不是它们的值。两个引用永远不会相等的机会很好。虽然这种弱点通常只会影响程序的正确性，但如果将相等性用于安全性决策，则可以利用它来影响程序的安全性。

### Extended Description:

In Java, using == or != to compare two strings for equality actually compares two objects for equality, not their values. Chances are good that the two references will never be equal. While this weakness often only affects program correctness, if the equality is used for a security decision, it could be leveraged to affect program security.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 598

提交日期 2008-04-11---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Information Exposure Through Query Strings in GET Request zh: -->GET请求中通过查询字符串的信息暴露*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序使用GET方法处理包含敏感信息的请求，这些信息可以通过浏览器的历史记录，引用，Web日志和其他来源公开该信息。

### Description:

The web application uses the GET method to process requests that contain sensitive information, which can expose that information through the browser's history, Referers, web logs, and other sources.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 599

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Missing Validation of OpenSSL Certificate zh: -->缺少对OpenSSL证书的验证*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用OpenSSL并信任或使用证书，而不使用SSL\_get\_verify\_result（）函数来确保证书满足所有必要的安全要求。

### Description:

The software uses OpenSSL and trusts or uses a certificate without using the SSL\_get\_verify\_result() function to ensure that the certificate satisfies all necessary security requirements.

### 详细描述:

这可能允许攻击者使用无效证书声称自己是可信主机，使用过期证书或进行其他攻击，如果证书得到正确验证，则可以检测到这些攻击。

### Extended Description:

This could allow an attacker to use an invalid certificate to claim to be a trusted host, use expired certificates, or conduct other attacks that could be detected if the certificate is properly validated.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

CWE-295和CWE-599非常相似，尽管CWE-599的范围更窄，仅适用于OpenSSL证书。因此，CWE-295的其他儿童也可被视为CWE-599的儿童。 CWE使用一维层次关系并不适合处理基于资源类型（“OpenSSL证书”作为“任何证书”的子项）和行为类型（“未验证到期”）等概念的不同类型的抽象关系“作为”不正确的验证“的孩子。

Weakness ID: 600

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Uncaught Exception in Servlet zh: -->Servlet中未捕获的异常*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Servlet不会捕获所有异常，这可能会泄露敏感的调试信息。

### Description:

The Servlet does not catch all exceptions, which may reveal sensitive debugging information.

### 详细描述:

当Servlet抛出异常时，Servlet容器发送回用户的默认错误响应通常包括调试信息。这些信息对攻击者来说非常有价值。例如，堆栈跟踪可能会向攻击者显示格式错误的SQL查询字符串，正在使用的数据库类型以及应用程序容器的版本。此信息使攻击者能够针对这些组件中的已知漏洞。

### Extended Description:

When a Servlet throws an exception, the default error response the Servlet container sends back to the user typically includes debugging information. This information is of great value to an attacker. For example, a stack trace might show the attacker a malformed SQL query string, the type of database being used, and the version of the application container. This information enables the attacker to target known vulnerabilities in these components.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

“Missing Catch Block”概念可能比Servlet更广泛，但CWE没有充分涵盖更广泛的概念。

Weakness ID: 601

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> URL Redirection to Untrusted Site ('Open Redirect') zh: -->URL重定向到不受信任的站点（'打开重定向'）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序接受用户控制的输入，该输入指定指向外部站点的链接，并在重定向中使用该链接。这简化了网络钓鱼攻击。

### Description:

A web application accepts a user-controlled input that specifies a link to an external site, and uses that link in a Redirect. This simplifies phishing attacks.

### 详细描述:

http参数可能包含URL值，并可能导致Web应用程序将请求重定向到指定的URL。通过将URL值修改为恶意站点，攻击者可能会成功启动网络钓鱼诈骗并窃取用户凭据。由于修改后的链接中的服务器名称与原始站点相同，因此网络钓鱼尝试具有更可靠的外观。

### Extended Description:

An http parameter may contain a URL value and could cause the web application to redirect the request to the specified URL. By modifying the URL value to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials. Because the server name in the modified link is identical to the original site, phishing attempts have a more trustworthy appearance.

### 问题背景 (Background Detail):

网络钓鱼是欺骗性企图强制用户进行身份盗用的私人信息的一般术语。

### 笔记 (Notes):

没有笔记

Weakness ID: 602

提交日期 2008-04-11---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Client-Side Enforcement of Server-Side Security zh: -->客户端执行服务器端安全性*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件由依赖客户端实现旨在保护服务器的机制的服务器组成。

### Description:

The software is composed of a server that relies on the client to implement a mechanism that is intended to protect the server.

### 详细描述:

当服务器依赖于客户端上的保护机制时，攻击者可以修改客户端行为以绕过保护机制，从而导致客户端和服务器之间可能发生意外交互。后果将根据机制试图保护的内容而有所不同。

### Extended Description:

When the server relies on protection mechanisms placed on the client side, an attacker can modify the client-side behavior to bypass the protection mechanisms resulting in potentially unexpected interactions between the client and server. The consequences will vary, depending on what the mechanisms are trying to protect.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

客户端安全的服务器端强制执行在概念上可能会发生，但某些体系结构可能具有这些强依赖性作为合法行为的一部分，例如瘦客户端。

Weakness ID: 603

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of Client-Side Authentication zh: -->使用客户端身份验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

客户端/服务器产品在客户端代码中执行身份验证，但不在服务器代码中执行身份验证，允许通过省略身份验证检查的已修改客户端绕过服务器端身份验证。

### Description:

A client/server product performs authentication within client code but not in server code, allowing server-side authentication to be bypassed via a modified client that omits the authentication check.

### 详细描述:

客户端身份验证非常弱，可能很容易被破坏。任何攻击者都可以阅读源代码并对身份验证机制进行反向工程，以访问本应受到保护的应用程序部分。

### Extended Description:

Client-side authentication is extremely weak and may be breached easily. Any attacker may read the source code and reverse-engineer the authentication mechanism to access parts of the application which would otherwise be protected.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

请注意，这一弱点与CWE-656（依赖安全的安全依赖）之间存在密切关系。如果开发人员不相信用户可以对客户端进行逆向工程，那么他们更有可能选择客户端身份验证，因为他们认为这是安全的。

Weakness ID: 605

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Multiple Binds to the Same Port zh: -->多个绑定到同一端口*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当允许多个套接字绑定到同一端口时，该端口上的其他服务可能被盗或被欺骗。

### Description:

When multiple sockets are allowed to bind to the same port, other services on that port may be stolen or spoofed.

### 详细描述:

在大多数系统上，设置SO\_REUSEADDR套接字选项和调用bind（）的组合允许任何进程绑定到先前进程与INADDR\_ANY绑定的端口。这允许用户绑定到非特权端口上绑定到INADDR\_ANY的服务器的特定地址，并窃取其UDP数据包/ TCP连接。

### Extended Description:

On most systems, a combination of setting the SO\_REUSEADDR socket option, and a call to bind() allows any process to bind to a port to which a previous process has bound with INADDR\_ANY. This allows a user to bind to the specific address of a server bound to INADDR\_ANY on an unprivileged port, and steal its UDP packets/TCP connection.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 606

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Unchecked Input for Loop Condition zh: -->未经检查的循环条件输入*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品未正确检查用于循环条件的输入，可能由于过度循环而导致拒绝服务。

### Description:

The product does not properly check inputs that are used for loop conditions, potentially leading to a denial of service because of excessive looping.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 607

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Public Static Final Field References Mutable Object zh: -->公共静态最终字段引用可变对象*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

公共或受保护的静态final字段引用可变对象，该对象允许通过恶意代码更改对象，或者意外地从另一个包更改对象。

### Description:

A public or protected static final field references a mutable object, which allows the object to be changed by malicious code, or accidentally from another package.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 608

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Struts: Non-private Field in ActionForm Class zh: -->Struts：ActionForm类中的非私有字段*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

ActionForm类包含一个尚未声明为私有的字段，可以在不使用setter或getter的情况下访问该字段。

### Description:

An ActionForm class contains a field that has not been declared private, which can be accessed without using a setter or getter.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 609

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Double-Checked Locking zh: -->双重锁定*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序使用双重检查锁定来访问资源，而没有显式同步的开销，但锁定不充分。

### Description:

The program uses double-checked locking to access a resource without the overhead of explicit synchronization, but the locking is insufficient.

### 详细描述:

双重检查锁定指的是程序员检查资源是否已初始化，抓取锁定，再次检查以查看资源是否已初始化的情况，然后执行初始化（如果尚未发生）。不应该这样做，因为不能保证在所有语言和所有体系结构中都能工作。总之，其他线程可能无法在同步块内运行，并且无法保证看到操作的执行顺序与它们在同步块中出现的顺序相同。

### Extended Description:

Double-checked locking refers to the situation where a programmer checks to see if a resource has been initialized, grabs a lock, checks again to see if the resource has been initialized, and then performs the initialization if it has not occurred yet. This should not be done, as is not guaranteed to work in all languages and on all architectures. In summary, other threads may not be operating inside the synchronous block and are not guaranteed to see the operations execute in the same order as they would appear inside the synchronous block.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 610

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Externally Controlled Reference to a Resource in Another Sphere zh: -->对另一个领域中的资源的外部控制引用*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用外部控制的名称或引用，该名称或引用解析为预期控制范围之外的资源。

### Description:

The product uses an externally controlled name or reference that resolves to a resource that is outside of the intended control sphere.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这是一个普遍的弱点，但大多数研究都集中在更专业的案例上，例如路径遍历（CWE-22）和符号链接（CWE-61）。符号链接有一个名称;通常，它看起来像文件系统中的任何其他文件。但是，该链接包括对另一个文件的引用，通常在另一个目录中 - 可能在另一个控制范围内。许多接受文件名的常见库函数将“跟随”符号链接并改为使用链接的目标。

Weakness ID: 611

提交日期 2011-03-29---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Restriction of XML External Entity Reference ('XXE') zh: -->XML外部实体引用的限制不当（'XXE'）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件处理一个XML文档，该文档可以包含带有URI的XML实体，这些URI可以解析为预期控制范围之外的文档，从而导致产品将不正确的文档嵌入其输出中。

### Description:

The software processes an XML document that can contain XML entities with URIs that resolve to documents outside of the intended sphere of control, causing the product to embed incorrect documents into its output.

### 详细描述:

XML文档可选地包含文档类型定义（Document Type Definition，DTD），除了其他功能之外，它还支持XML实体的定义。可以通过以URI的形式提供替换字符串来定义实体。 XML解析器可以访问此URI的内容，并将这些内容嵌入到XML文档中以供进一步处理。  
通过提交使用file：// URI定义外部实体的XML文件，攻击者可以使处理应用程序读取本地文件的内容。例如，诸如“file：/// c：/winnt/win.ini”之类的URI指定（在Windows中）文件C：\ Winnt \ win.ini，或者file：/// etc / passwd指定密码基于Unix的系统中的文件。将URI与其他方案（如http：//）一起使用，攻击者可以强制应用程序向攻击者无法直接访问的服务器发出传出请求，这可以用来绕过防火墙限制或隐藏端口扫描等攻击源。  
读取URI的内容后，它将反馈到正在处理XML的应用程序中。该应用程序可以回送数据（例如，在错误消息中），从而暴露文件内容。

### Extended Description:

XML documents optionally contain a Document Type Definition (DTD), which, among other features, enables the definition of XML entities. It is possible to define an entity by providing a substitution string in the form of a URI. The XML parser can access the contents of this URI and embed these contents back into the XML document for further processing.  
By submitting an XML file that defines an external entity with a file:// URI, an attacker can cause the processing application to read the contents of a local file. For example, a URI such as "file:///c:/winnt/win.ini" designates (in Windows) the file C:\Winnt\win.ini, or file:///etc/passwd designates the password file in Unix-based systems. Using URIs with other schemes such as http://, the attacker can force the application to make outgoing requests to servers that the attacker cannot reach directly, which can be used to bypass firewall restrictions or hide the source of attacks such as port scanning.  
Once the content of the URI is read, it is fed back into the application that is processing the XML. This application may echo back the data (e.g. in an error message), thereby exposing the file contents.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

CWE-918（SSRF）和CWE-611（XXE）密切相关，因为它们都涉及与Web相关的技术，并且可以向意外目的地发出出站请求。但是，XXE可以在客户端执行，或者在软件不直接作为服务器的其他环境中执行，因此SSRF首字母缩略词的“服务器”部分不一定适用。

Weakness ID: 612

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Indexing of Private Data zh: -->通过索引私有数据进行信息曝光*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品针对私有文档执行索引例程，但没有充分验证可以访问索引的actor还具有访问私有文档的权限。

### Description:

The product performs an indexing routine against private documents, but does not sufficiently verify that the actors who can access the index also have the privileges to access the private documents.

### 详细描述:

当针对一组私有文档应用索引例程，并且该索引的结果可供无法访问这些文档的外部人员使用时，外部人员可能能够通过进行有针对性的搜索来获取敏感信息。如果搜索结果包含不属于搜索查询的周围文本，则风险尤其危险。此问题可能出现在未配置（或实现）的搜索引擎中，以忽略应保持隐藏的关键文件;即使没有直接下载这些文件的权限，远程用户也可以阅读它们。

### Extended Description:

When an indexing routine is applied against a group of private documents, and that index's results are available to outsiders who do not have access to those documents, then outsiders might be able to obtain sensitive information by conducting targeted searches. The risk is especially dangerous if search results include surrounding text that was not part of the search query. This issue can appear in search engines that are not configured (or implemented) to ignore critical files that should remain hidden; even without permissions to download these files directly, the remote user could read them.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这种弱点可能未得到充分研究和报道不足

Weakness ID: 613

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Insufficient Session Expiration zh: -->会话过期不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

据WASC称，“会话过期不足是指网站允许攻击者重用旧会话凭证或会话ID进行授权。”

### Description:

According to WASC, "Insufficient Session Expiration is when a web site permits an attacker to reuse old session credentials or session IDs for authorization."

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

缺乏适当的会话到期可能会改善某些攻击的可能成功。例如，攻击者可能通过网络嗅探器或跨站点脚本攻击拦截会话ID。虽然如果立即使用被盗令牌，短会话到期时间无效，但它们将防止正在进行的会话ID重播。在另一种情况下，用户可以从共享计算机（例如在图书馆，网吧或开放工作环境）访问网站。会话过期不足可能允许攻击者使用浏览器的后退按钮访问受害者先前访问过的网页。

Weakness ID: 614

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Sensitive Cookie in HTTPS Session Without 'Secure' Attribute zh: -->没有“安全”属性的HTTPS会话中的敏感Cookie*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

未设置HTTPS会话中敏感cookie的安全属性，这可能导致用户代理通过HTTP会话以纯文本格式发送这些cookie。

### Description:

The Secure attribute for sensitive cookies in HTTPS sessions is not set, which could cause the user agent to send those cookies in plaintext over an HTTP session.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 615

提交日期 2011-03-29---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through Comments zh: -->信息曝光评论*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

虽然添加一般注释非常有用，但是一些程序员倾向于保留重要数据，例如：与Web应用程序相关的文件名，旧链接或用户不想浏览的链接，旧代码片段等。

### Description:

While adding general comments is very useful, some programmers tend to leave important data, such as: filenames related to the web application, old links or links which were not meant to be browsed by users, old code fragments, etc.

### 详细描述:

发现这些注释的攻击者可以映射应用程序的结构和文件，公开站点的隐藏部分，并研究代码片段以对应用程序进行反向工程，这可能有助于对站点进行进一步的攻击。

### Extended Description:

An attacker who finds these comments can map the application's structure and files, expose hidden parts of the site, and study the fragments of code to reverse engineer the application, which may help develop further attacks against the site.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 616

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Incomplete Identification of Uploaded File Variables (PHP) zh: -->上传文件变量的不完整标识（PHP）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

PHP应用程序使用旧方法处理上传文件，方法是引用为每个文件设置的四个全局变量（例如$ varname，$ varname\_size，$ varname\_name，$ varname\_type）。这些变量可能被攻击者覆盖，导致应用程序处理未经授权的文件。

### Description:

The PHP application uses an old method for processing uploaded files by referencing the four global variables that are set for each file (e.g. $varname, $varname\_size, $varname\_name, $varname\_type). These variables could be overwritten by attackers, causing the application to process unauthorized files.

### 详细描述:

POST请求，cookie或其他填充或覆盖这些变量的方法可能会覆盖这些全局变量。这可以通过提供诸如“/ etc / passwd”之类的值来读取或处理任意文件。

### Extended Description:

These global variables could be overwritten by POST requests, cookies, or other methods of populating or overwriting these variables. This could be used to read or process arbitrary files by providing values such as "/etc/passwd".

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 617

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Reachable Assertion zh: -->可达到的断言*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品包含一个可由攻击者触发的assert（）或类似语句，这会导致应用程序退出或其他比必要更严重的行为。

### Description:

The product contains an assert() or similar statement that can be triggered by an attacker, which leads to an application exit or other behavior that is more severe than necessary.

### 详细描述:

虽然断言有助于捕获逻辑错误并降低达到更严重漏洞条件的可能性，但它仍然可能导致拒绝服务。  
例如，如果服务器处理多个并发连接，并且在一个连接中发生assert（）导致所有其他连接被删除，则这是一个可达的断言，导致拒绝服务。

### Extended Description:

While assertion is good for catching logic errors and reducing the chances of reaching more serious vulnerability conditions, it can still lead to a denial of service.  
For example, if a server handles multiple simultaneous connections, and an assert() occurs in one single connection that causes all other connections to be dropped, this is a reachable assertion that leads to a denial of service.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 618

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Exposed Unsafe ActiveX Method zh: -->暴露的不安全的ActiveX方法*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

ActiveX控件旨在用于Web浏览器，但它暴露了执行浏览器安全模型之外的操作的危险方法（例如区域或域）。

### Description:

An ActiveX control is intended for use in a web browser, but it exposes dangerous methods that perform actions that are outside of the browser's security model (e.g. the zone or domain).

### 详细描述:

与典型的Java或javascript相比，ActiveX控件可以对操作系统进行更大的控制。暴露的方法可能会受到各种漏洞的影响，具体取决于这些方法的实现行为，以及是否对提供的参数执行输入验证。如果没有完整性检查或原始验证，攻击者可以调用此方法。

### Extended Description:

ActiveX controls can exercise far greater control over the operating system than typical Java or javascript. Exposed methods can be subject to various vulnerabilities, depending on the implemented behaviors of those methods, and whether input validation is performed on the provided arguments. If there is no integrity checking or origin validation, this method could be invoked by attackers.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 619

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Dangling Database Cursor ('Cursor Injection') zh: -->悬空数据库光标（'光标注入'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果未正确关闭数据库游标，则其他用户可以访问它，同时保留最初分配的相同权限，将游标保留为“悬空”。

### Description:

If a database cursor is not closed properly, then it could become accessible to other users while retaining the same privileges that were originally assigned, leaving the cursor "dangling."

### 详细描述:

例如，未处理的异常可能会产生不正确的悬空游标。问题的影响取决于游标的角色，但SQL注入攻击通常是可能的。

### Extended Description:

For example, an improper dangling cursor could arise from unhandled exceptions. The impact of the issue depends on the cursor's role, but SQL injection attacks are commonly possible.

### 问题背景 (Background Detail):

游标是Oracle PL / SQL和其他语言中的一项功能，它提供了执行和访问SQL查询结果的句柄。

### 笔记 (Notes):

没有笔记

Weakness ID: 620

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Unverified Password Change zh: -->未验证的密码更改*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

为用户设置新密码时，产品不需要了解原始密码或使用其他形式的身份验证。

### Description:

When setting a new password for a user, the product does not require knowledge of the original password, or using another form of authentication.

### 详细描述:

攻击者可以使用此方法更改其他用户的密码，从而获得与该用户关联的权限。

### Extended Description:

This could be used by an attacker to change passwords for another user, thus gaining the privileges associated with that user.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 621

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Variable Extraction Error zh: -->变量提取错误*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用外部输入来确定提取信息的变量的名称，而不验证指定变量的名称是否有效。这可能导致程序覆盖非预期的变量。

### Description:

The product uses external input to determine the names of variables into which information is extracted, without verifying that the names of the specified variables are valid. This could cause the program to overwrite unintended variables.

### 详细描述:

例如，在PHP中，提取可用于提供类似于register\_globals的功能，这是一种在生产系统中经常禁用的危险功能。在没有正确参数的情况下调用extract（）或import\_request\_variables（）可能会覆盖任意全局变量，包括超全局变量。  
其他解释语言也可以使用类似的功能，包括自定义语言。

### Extended Description:

For example, in PHP, extraction can be used to provide functionality similar to register\_globals, a dangerous functionality that is frequently disabled in production systems. Calling extract() or import\_request\_variables() without the proper arguments could allow arbitrary global variables to be overwritten, including superglobals.  
Similar functionality is possible in other interpreted languages, including custom languages.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

PHP可能报告不足。对其他解释语言的研究不足。

Weakness ID: 622

提交日期 2012-10-30---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Validation of Function Hook Arguments zh: -->函数钩子参数的不正确验证*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品会向用户可访问的API函数添加挂钩，但不会正确验证参数。这可能会导致漏洞。

### Description:

A product adds hooks to user-accessible API functions, but does not properly validate the arguments. This could lead to resultant vulnerabilities.

### 详细描述:

此类挂钩可用于以特权运行的防御软件，例如反病毒或防火墙，它可以挂钩内核调用。当参数未经验证时，它们可用于绕过保护方案或攻击产品本身。

### Extended Description:

Such hooks can be used in defensive software that runs with privileges, such as anti-virus or firewall, which hooks kernel calls. When the arguments are not validated, they could be used to bypass the protection scheme or attack the product itself.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 623

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Unsafe ActiveX Control Marked Safe For Scripting zh: -->不安全的ActiveX控件标记为脚本安全*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

ActiveX控件旨在限制使用，但已将其标记为可安全执行脚本。

### Description:

An ActiveX control is intended for restricted use, but it has been marked as safe-for-scripting.

### 详细描述:

这可能允许攻击者通过访问控件的网页使用危险功能，这可能导致不同的结果漏洞，具体取决于控件的行为。

### Extended Description:

This might allow attackers to use dangerous functionality via a web page that accesses the control, which can lead to different resultant vulnerabilities, depending on the control's behavior.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

怀疑这是报告不足。

Weakness ID: 624

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Executable Regular Expression Error zh: -->可执行的正则表达式错误*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用正则表达式，或者（1）包含具有用户控制输入的可执行组件，或者（2）允许用户通过插入模式修饰符来启用执行。

### Description:

The product uses a regular expression that either (1) contains an executable component with user-controlled inputs, or (2) allows a user to enable execution by inserting pattern modifiers.

### 详细描述:

可以在PHP preg\_replace（）函数中使用Case（2），并且可能在其他语言中将用户控制的输入插入到稍后被解析为正则表达式的字符串中。

### Extended Description:

Case (2) is possible in the PHP preg\_replace() function, and possibly in other languages when a user-controlled input is inserted into a string that is later parsed as a regular expression.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。现有的PHP报告仅限于技术娴熟的研究人员，但其他语言的例子很少。怀疑所有语言的报道都不足。可用性因素可能使其在PHP中更为普遍，但该理论尚未得到研究。

Weakness ID: 625

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Permissive Regular Expression zh: -->允许正则表达式*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用的正则表达式不足以限制允许值的集合。

### Description:

The product uses a regular expression that does not sufficiently restrict the set of allowed values.

### 详细描述:

这有效地使正则表达式接受与模式匹配的子串，从而产生与目标的部分比较。在某些情况下，这可能会导致其他弱点。常见错误包括：  
  
  
不识别目标字符串的开头和结尾  
使用通配符而不是可接受的字符范围  
其他

### Extended Description:

This effectively causes the regexp to accept substrings that match the pattern, which produces a partial comparison to the target. In some cases, this can lead to other weaknesses. Common errors include:  
  
  
not identifying the beginning and end of the target string  
using wildcards instead of acceptable character ranges  
others

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 626

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Null Byte Interaction Error (Poison Null Byte) zh: -->空字节交互错误（Poison Null Byte）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在不同表示或组件之间传递数据时，产品无法正确处理空字节或NUL字符。

### Description:

The product does not properly handle null bytes or NUL characters when passing data between different representations or components.

### 详细描述:

空字节（NUL字符）可以在表示或语言之间具有不同的含义。例如，它是标准C库中的字符串终止符，但Perl和PHP字符串不会将其视为终止符。当两个表示相交时 - 例如当Perl或PHP调用底层C功能时 - 这会产生具有意外结果的交互错误。 ASP已经报道了类似的问题。用C编写的其他口译员也可能受到影响。  
通过终止添加到文件名的硬编码扩展，毒性空字节在路径遍历攻击中经常有用。它可以在PHP中的正则表达式处理中发挥作用。

### Extended Description:

A null byte (NUL character) can have different meanings across representations or languages. For example, it is a string terminator in standard C libraries, but Perl and PHP strings do not treat it as a terminator. When two representations are crossed - such as when Perl or PHP invokes underlying C functionality - this can produce an interaction error with unexpected results. Similar issues have been reported for ASP. Other interpreters written in C might also be affected.  
The poison null byte is frequently useful in path traversal attacks by terminating hard-coded extensions that are added to a filename. It can play a role in regular expression processing in PHP.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

“毒性空字节”的当前用法通常与此C / Perl / PHP交互错误有关，但1998年的原始术语适用于涉及空字节的逐个缓冲区溢出。

Weakness ID: 627

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Dynamic Variable Evaluation zh: -->动态变量评估*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在用户可以在运行时影响变量名称的语言中，如果不控制变量名，攻击者可以读取或写入任意变量，或访问任意函数。

### Description:

In a language where the user can influence the name of a variable at runtime, if the variable names are not controlled, an attacker can read or write to arbitrary variables, or access arbitrary functions.

### 详细描述:

由此产生的漏洞取决于应用程序的行为，无论是在交叉点还是在相关变量或函数可访问的任何控制/数据流中。

### Extended Description:

The resultant vulnerabilities depend on the behavior of the application, both at the crossover point and in any control/data flow that is reachable by the related variables or functions.

### 问题背景 (Background Detail):

许多解释语言都支持使用“$$ varname”构造来设置名称由$ varname变量指定的变量。在PHP中，这些被称为“变量变量”。也可以使用类似的语法调用函数，例如$$ funcname（arg1，arg2）。

### 笔记 (Notes):

研究不足，可能报道不足。很少有研究人员在寻找这个问题;尽管其他语言受到影响，但大多数公开报告都是针对PHP的。随着开发人员开始实现功能而不是register\_globals，这个问题可能会在PHP中增长。

Weakness ID: 628

提交日期 2008-04-11---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Function Call with Incorrectly Specified Arguments zh: -->函数调用具有错误指定的参数*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用未正确指定的参数调用函数，过程或例程，从而导致始终不正确的行为和由此产生的弱点。

### Description:

The product calls a function, procedure, or routine with arguments that are not correctly specified, leading to always-incorrect behavior and resultant weaknesses.

### 详细描述:

有多种方法可以引入这种弱点，包括：  
  
  
错误的变量或参考;  
参数数量不正确;  
参数的顺序不正确;  
错误的论点类型;要么  
错误的价值。

### Extended Description:

There are multiple ways in which this weakness can be introduced, including:  
  
  
the wrong variable or reference;  
an incorrect number of arguments;  
incorrect order of arguments;  
wrong type of arguments; or  
wrong value.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 636

提交日期 2008-09-09---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Not Failing Securely ('Failing Open') zh: -->没有安全失败（'失败打开'）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当产品遇到错误情况或故障时，其设计要求它回退到比其他可用选项安全性低的状态，例如选择最弱的加密算法或使用最宽松的访问控制限制。

### Description:

When the product encounters an error condition or failure, its design requires it to fall back to a state that is less secure than other options that are available, such as selecting the weakest encryption algorithm or using the most permissive access control restrictions.

### 详细描述:

通过进入不太安全的状态，产品继承了与该状态相关的弱点，使其更容易妥协。至少，它会导致管理员产生错误的安全感。这种弱点通常是由于希望“功能失效”以最小化管理和支持成本而不是“失败安全”。

### Extended Description:

By entering a less secure state, the product inherits the weaknesses associated with that state, making it easier to compromise. At the least, it causes administrators to have a false sense of security. This weakness typically occurs as a result of wanting to "fail functional" to minimize administration and support costs, instead of "failing safe."

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

由于设计问题难以修复，因此很少公开报告，因此截至2008年1月，此问题的CVE示例很少。大多数公开报告的问题是由于实施错误而非设计而发生的，例如CVE-2005- 3177（大量资源处理不当）或CVE-2005-2969（无意中禁用验证步骤，导致选择较弱的协议）。

Weakness ID: 637

提交日期 2009-01-12---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Unnecessary Complexity in Protection Mechanism (Not Using 'Economy of Mechanism') zh: -->保护机制中不必要的复杂性（不使用'机制经济'）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用了比必要的更复杂的机制，当机制未被正确理解，建模，配置，实现或使用时，这可能导致最终的弱点。

### Description:

The software uses a more complex mechanism than necessary, which could lead to resultant weaknesses when the mechanism is not correctly understood, modeled, configured, implemented, or used.

### 详细描述:

安全机制应尽可能简单。复杂的安全机制可能会导致部分实现和兼容性问题，从而导致假设和实现的安全性不匹配。这个原则的必然结果是数据规范应该尽可能简单，因为复杂的数据规范会导致复杂的验证代码。复杂的任务和系统也可能需要通过复杂的安全检查来保护，因此应该首选简单的系统。

### Extended Description:

Security mechanisms should be as simple as possible. Complex security mechanisms may engender partial implementations and compatibility problems, with resulting mismatches in assumptions and implemented security. A corollary of this principle is that data specifications should be as simple as possible, because complex data specifications result in complex validation code. Complex tasks and systems may also need to be guarded by complex security checks, so simple systems should be preferred.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 638

提交日期 2009-01-12---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Not Using Complete Mediation zh: -->不使用完整调解*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

每次实体访问资源时，软件都不会对资源执行访问检查，如果该实体的权限或特权随时间发生变化，则可能会产生缺陷。

### Description:

The software does not perform access checks on a resource every time the resource is accessed by an entity, which can create resultant weaknesses if that entity's rights or privileges change over time.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 639

提交日期 2011-03-29---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Authorization Bypass Through User-Controlled Key zh: -->授权绕过用户控制的密钥*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

系统的授权功能不会阻止一个用户通过修改标识数据的密钥值来访问另一个用户的数据或记录。

### Description:

The system's authorization functionality does not prevent one user from gaining access to another user's data or record by modifying the key value identifying the data.

### 详细描述:

基于用户控制下的某个键值在系统中进行用户记录的检索。密钥通常将识别存储在系统中的用户相关记录，并且将用于查找该记录以呈现给用户。攻击者可能必须是系统中经过身份验证的用户。但是，授权过程不会正确检查数据访问操作，以确保执行操作的经过身份验证的用户具有足够的权限来执行所请求的数据访问，从而绕过系统中存在的任何其他授权检查。  
例如，攻击者可以查看检索用户特定数据的地方（例如搜索屏幕）并确定正在查找的项目的密钥是否可从外部控制。键可以是HTML表单字段中的隐藏字段，可以作为URL参数或未加密的cookie变量传递，然后在每种情况下都可以篡改键值。  
这种弱点的一个表现是系统使用顺序或其他易于猜测的会话ID，这将允许一个用户轻松切换到另一个用户的会话并读取/修改他们的数据。

### Extended Description:

Retrieval of a user record occurs in the system based on some key value that is under user control. The key would typically identify a user-related record stored in the system and would be used to lookup that record for presentation to the user. It is likely that an attacker would have to be an authenticated user in the system. However, the authorization process would not properly check the data access operation to ensure that the authenticated user performing the operation has sufficient entitlements to perform the requested data access, hence bypassing any other authorization checks present in the system.  
For example, attackers can look at places where user specific data is retrieved (e.g. search screens) and determine whether the key for the item being looked up is controllable externally. The key may be a hidden field in the HTML form field, might be passed as a URL parameter or as an unencrypted cookie variable, then in each of these cases it will be possible to tamper with the key value.  
One manifestation of this weakness is when a system uses sequential or otherwise easily-guessable session IDs that would allow one user to easily switch to another user's session and read/modify their data.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 640

提交日期 2008-09-09---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Weak Password Recovery Mechanism for Forgotten Password zh: -->忘记密码的弱密码恢复机制*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一种机制，用户可以在不知道原始密码的情况下恢复或更改密码，但机制很弱。

### Description:

The software contains a mechanism for users to recover or change their passwords without knowing the original password, but the mechanism is weak.

### 详细描述:

应用程序通常具有一种机制，该机制为用户在忘记密码时提供访问其帐户的权限。通常，密码恢复机制很弱，其结果是使得合法系统用户以外的人更有可能获得对该用户帐户的访问权。弱密码恢复方案完全破坏了强密码认证方案。  
这个弱点可能是安全问题太容易猜测或找不到答案（例如，因为问题太常见，或者可以使用社交媒体找到答案）。或者密码恢复机制代码中可能存在实现弱点，例如可能欺骗系统将新密码通过电子邮件发送到除用户之外的电子邮件帐户。密码重置率可能没有限制，因此如果攻击者试图快速连续恢复密码，则合法用户可能会被攻击者拒绝服务。系统可以将原始密码发送给用户，而不是生成新的临时密码。总之，密码恢复功能，如果不经过精心设计和实施，通常会成为系统中最薄弱的环节，可能会以允许攻击者未经授权访问系统的方式被滥用。

### Extended Description:

It is common for an application to have a mechanism that provides a means for a user to gain access to their account in the event they forget their password. Very often the password recovery mechanism is weak, which has the effect of making it more likely that it would be possible for a person other than the legitimate system user to gain access to that user's account. Weak password recovery schemes completely undermine a strong password authentication scheme.  
This weakness may be that the security question is too easy to guess or find an answer to (e.g. because the question is too common, or the answers can be found using social media). Or there might be an implementation weakness in the password recovery mechanism code that may for instance trick the system into e-mailing the new password to an e-mail account other than that of the user. There might be no throttling done on the rate of password resets so that a legitimate user can be denied service by an attacker if an attacker tries to recover their password in a rapid succession. The system may send the original password to the user rather than generating a new temporary password. In summary, password recovery functionality, if not carefully designed and implemented can often become the system's weakest link that can be misused in a way that would allow an attacker to gain unauthorized access to the system.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目可能会重新分类为类别或“松散复合”，因为它列出了可能使机制变弱的多个特定错误。但是，在视图1000下，它可能是保护机制失败的弱点，尽管它与大多数PMF问题不同，因为它与旨在绕过保护机制的功能有关（具体地说，缺乏对密码的了解） 。

Weakness ID: 641

提交日期 2010-06-21---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Restriction of Names for Files and Other Resources zh: -->文件和其他资源的名称限制不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序使用来自上游组件的输入构造文件或其他资源的名称，但它不限制或不正确地限制结果名称。

### Description:

The application constructs the name of a file or other resource using input from an upstream component, but it does not restrict or incorrectly restricts the resulting name.

### 详细描述:

这可能会产生缺陷。例如，如果这些资源的名称包含脚本字符，则如果应用程序曾在动态生成的网页上显示资源的名称，则可能会在客户端的浏览器中执行脚本。或者，如果某些应用程序解析器使用了资源，则特制名称可能会利用解析器内部的某些漏洞，从而可能导致在服务器计算机上执行任意代码。问题将根据这些格式错误的资源名称的使用情况以及是否存在漏洞或由目标技术做出的假设而有所不同，这些技术将使代码执行成为可能。

### Extended Description:

This may produce resultant weaknesses. For instance, if the names of these resources contain scripting characters, it is possible that a script may get executed in the client's browser if the application ever displays the name of the resource on a dynamically generated web page. Alternately, if the resources are consumed by some application parser, a specially crafted name can exploit some vulnerability internal to the parser, potentially resulting in execution of arbitrary code on the server machine. The problems will vary based on the context of usage of such malformed resource names and whether vulnerabilities are present in or assumptions are made by the targeted technology that would make code execution possible.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 642

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> External Control of Critical State Data zh: -->关键状态数据的外部控制*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件将有关其用户或软件本身的安全关键状态信息存储在未经授权的参与者可访问的位置。

### Description:

The software stores security-critical state information about its users, or the software itself, in a location that is accessible to unauthorized actors.

### 详细描述:

如果攻击者可以在未经检测的情况下修改状态信息，则可以使用它来执行未经授权的操作或访问意外资源，因为应用程序员不期望可以更改状态。  
状态信息可以存储在各种位置，例如cookie，隐藏的Web表单字段，输入参数或参数，环境变量，数据库记录，设置文件中等。所有这些位置都有可能被修改由攻击者当此状态信息用于控制安全性或确定资源使用情况时，它可能会创建漏洞。例如，应用程序可以执行身份验证，然后将状态保存在“authenticated = true”cookie中。攻击者可能只是创建此cookie以绕过身份验证。

### Extended Description:

If an attacker can modify the state information without detection, then it could be used to perform unauthorized actions or access unexpected resources, since the application programmer does not expect that the state can be changed.  
State information can be stored in various locations such as a cookie, in a hidden web form field, input parameter or argument, an environment variable, a database record, within a settings file, etc. All of these locations have the potential to be modified by an attacker. When this state information is used to control security or determine resource usage, then it may create a vulnerability. For example, an application may perform authentication, then save the state in an "authenticated=true" cookie. An attacker may simply create this cookie in order to bypass the authentication.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 643

提交日期 2008-10-14---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Neutralization of Data within XPath Expressions ('XPath Injection') zh: -->XPath表达式中的数据中和不正确（'XPath注入'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入动态构造用于从XML数据库检索数据的XPath表达式，但它不会中和或错误地中和该输入。这允许攻击者控制查询的结构。

### Description:

The software uses external input to dynamically construct an XPath expression used to retrieve data from an XML database, but it does not neutralize or incorrectly neutralizes that input. This allows an attacker to control the structure of the query.

### 详细描述:

实际效果是攻击者可以控制从XML数据库中选择的信息，并可以使用该功能来控制应用程序流，修改逻辑，检索未经授权的数据或绕过重要检查（例如身份验证）。

### Extended Description:

The net effect is that the attacker will have control over the information selected from the XML database and may use that ability to control application flow, modify logic, retrieve unauthorized data, or bypass important checks (e.g. authentication).

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这种弱点类似于启用注入式攻击的其他弱点，例如SQL注入，命令注入和LDAP注入。主要区别在于此处的攻击目标是XML数据库。

Weakness ID: 644

提交日期 2008-10-14---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Neutralization of HTTP Headers for Scripting Syntax zh: -->脚本语法的HTTP标头中和不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该应用程序不会中和或错误地中和HTTP标头中的Web脚本语法，可以由可以处理原始标头的Web浏览器组件（如Flash）使用。

### Description:

The application does not neutralize or incorrectly neutralizes web scripting syntax in HTTP headers that can be used by web browser components that can process raw headers, such as Flash.

### 详细描述:

攻击者可能能够对启用了这些组件的用户进行跨站点脚本攻击和其他攻击。  
如果应用程序不中和放置在来自服务器的HTTP响应的标头中的用户控制的数据，则标头可能包含将在客户端的浏览器上下文中执行的脚本，可能导致跨站点脚本漏洞或可能导致HTTP响应分裂攻击。仔细控制HTTP响应头和HTTP响应体中放置的数据非常重要，以确保不存在脚本语法，并考虑各种编码。

### Extended Description:

An attacker may be able to conduct cross-site scripting and other attacks against users who have these components enabled.  
If an application does not neutralize user controlled data being placed in the header of an HTTP response coming from the server, the header may contain a script that will get executed in the client's browser context, potentially resulting in a cross site scripting vulnerability or possibly an HTTP response splitting attack. It is important to carefully control data that is being placed both in HTTP response header and in the HTTP response body to ensure that no scripting syntax is present, taking various encodings into account.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 645

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Overly Restrictive Account Lockout Mechanism zh: -->过度限制性账户锁定机制*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个帐户锁定保护机制，但该机制限制太多，可以轻松触发，这允许攻击者通过导致其帐户被锁定来拒绝为合法用户提供服务。

### Description:

The software contains an account lockout protection mechanism, but the mechanism is too restrictive and can be triggered too easily, which allows attackers to deny service to legitimate users by causing their accounts to be locked out.

### 详细描述:

帐户锁定是应用程序中经常出现的安全功能，作为对基于密码的系统身份验证机制的暴力攻击的对策。在一定次数的登录尝试失败后，用户的帐户可能会被禁用一段时间或者直到管理员解锁。其他安全事件也可能触发帐户锁定。但是，攻击者可能会使用此安全功能拒绝向合法系统用户提供服务。因此，确保帐户锁定安全机制不会过于严格，这一点很重要。

### Extended Description:

Account lockout is a security feature often present in applications as a countermeasure to the brute force attack on the password based authentication mechanism of the system. After a certain number of failed login attempts, the users' account may be disabled for a certain period of time or until it is unlocked by an administrator. Other security events may also possibly trigger account lockout. However, an attacker may use this very security feature to deny service to legitimate system users. It is therefore important to ensure that the account lockout security mechanism is not overly restrictive.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 646

提交日期 2008-10-14---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Reliance on File Name or Extension of Externally-Supplied File zh: -->依赖文件名或外部提供文件的扩展*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件允许上载文件，但它依赖于文件名或文件扩展名来确定适当的行为。攻击者可以使用此方法使文件被错误分类并以危险的方式处理。

### Description:

The software allows a file to be uploaded, but it relies on the file name or extension of the file to determine the appropriate behaviors. This could be used by attackers to cause the file to be misclassified and processed in a dangerous fashion.

### 详细描述:

应用程序可能会使用用户提供的文件的文件名或扩展名来确定正确的操作过程，例如选择应该传递控制权的正确进程，决定应该提供哪些数据，或者应该使用哪些资源分配。如果攻击者可能导致代码错误地分类提供的文件，则可能会发生错误的操作。例如，攻击者可以提供以“.php.gif”扩展名结尾的文件，该扩展名似乎是GIF图像，但将作为PHP代码处理。在极端情况下，代码执行是可能的，但攻击者也可能导致资源耗尽，拒绝服务，调试或系统数据（包括应用程序源代码）的暴露，或绑定到特定的服务器端进程。这种弱点可能是由于Web和应用程序服务器使用的任何技术中的漏洞，由于配置错误或应用程序本身的另一个漏洞造成的。

### Extended Description:

An application might use the file name or extension of of a user-supplied file to determine the proper course of action, such as selecting the correct process to which control should be passed, deciding what data should be made available, or what resources should be allocated. If the attacker can cause the code to misclassify the supplied file, then the wrong action could occur. For example, an attacker could supply a file that ends in a ".php.gif" extension that appears to be a GIF image, but would be processed as PHP code. In extreme cases, code execution is possible, but the attacker could also cause exhaustion of resources, denial of service, exposure of debug or system data (including application source code), or being bound to a particular server side process. This weakness may be due to a vulnerability in any of the technologies used by the web and application servers, due to misconfiguration, or resultant from another flaw in the application itself.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 647

提交日期 2008-10-14---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Non-Canonical URL Paths for Authorization Decisions zh: -->使用非规范URL路径进行授权决策*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件定义策略命名空间，并根据URL是规范的假设做出授权决策。这可以允许非规范URL绕过授权。

### Description:

The software defines policy namespaces and makes authorization decisions based on the assumption that a URL is canonical. This can allow a non-canonical URL to bypass the authorization.

### 详细描述:

如果应用程序定义策略命名空间并根据URL进行授权决策，但在做出授权决策之前不需要或转换为规范URL，则会打开应用程序进行攻击。例如，如果应用程序仅允许访问http://www.example.com/mypage，则攻击者可能可以使用等效的URL绕过此限制，例如：  
  
  
http://WWW.EXAMPLE.COM/mypage  
http://www.example.com/%6Dypage（备用编码）  
http://192.168.1.1/mypage（IP地址）  
http://www.example.com/mypage/（尾随/）  
http://www.example.com:80/mypage  
  
  
因此，重要的是指定基于路径信息的访问控制策略，其中所有备用编码都被拒绝（这可以通过默认拒绝规则来完成）。

### Extended Description:

If an application defines policy namespaces and makes authorization decisions based on the URL, but it does not require or convert to a canonical URL before making the authorization decision, then it opens the application to attack. For example, if the application only wants to allow access to http://www.example.com/mypage, then the attacker might be able to bypass this restriction using equivalent URLs such as:  
  
  
http://WWW.EXAMPLE.COM/mypage  
http://www.example.com/%6Dypage (alternate encoding)  
http://192.168.1.1/mypage (IP address)  
http://www.example.com/mypage/ (trailing /)  
http://www.example.com:80/mypage  
  
  
Therefore it is important to specify access control policy that is based on the path information in some canonical form with all alternate encodings rejected (which can be accomplished by a default deny rule).

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 648

提交日期 2009-05-27---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Incorrect Use of Privileged APIs zh: -->不正确使用特权API*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该应用程序不符合需要额外权限的函数调用的API要求。这可能允许攻击者通过错误地调用函数来获得权限。

### Description:

The application does not conform to the API requirements for a function call that requires extra privileges. This could allow attackers to gain privileges by causing the function to be called incorrectly.

### 详细描述:

当应用程序包含执行需要提升级别权限的操作的某些功能时，特权API的调用者必须小心：  
  
  
确保API所做的假设是有效的，例如参数的有效性  
说明API设计/实施中的已知缺陷  
从安全的上下文中调用API  
  
  
如果API的调用者不遵循这些要求，则可能允许恶意用户或进程提升其权限，劫持该进程或窃取敏感数据。  
例如，重要的是要知道特权API在返回调用者之前是否没有脱离其特权，或者特权函数是否可以对调用者传递给它的数据，上下文或状态信息做出某些假设。始终知道何时以及如何调用特权API以确保无法利用其提升的特权级别非常重要。

### Extended Description:

When an application contains certain functions that perform operations requiring an elevated level of privilege, the caller of a privileged API must be careful to:  
  
  
ensure that assumptions made by the APIs are valid, such as validity of arguments  
account for known weaknesses in the design/implementation of the API  
call the API from a safe context  
  
  
If the caller of the API does not follow these requirements, then it may allow a malicious user or process to elevate their privilege, hijack the process, or steal sensitive data.  
For instance, it is important to know if privileged APIs do not shed their privileges before returning to the caller or if the privileged function might make certain assumptions about the data, context or state information passed to it by the caller. It is important to always know when and how privileged APIs can be called in order to ensure that their elevated level of privilege cannot be exploited.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 649

提交日期 2008-04-11---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Reliance on Obfuscation or Encryption of Security-Relevant Inputs without Integrity Checking zh: -->在没有完整性检查的情况下依赖于对安全相关输入进行模糊处理或加密*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部参与者不应该可变的输入的模糊处理或加密，但软件不使用完整性检查来检测这些输入是否已被修改。

### Description:

The software uses obfuscation or encryption of inputs that should not be mutable by an external actor, but the software does not use integrity checks to detect if those inputs have been modified.

### 详细描述:

当应用程序依赖于混淆或错误应用/弱加密来保护客户端可控制的令牌或参数时，这可能会对用户状态，系统状态或服务器上的某些决策产生影响。在不保护令牌/参数的完整性的情况下，应用程序易受攻击的攻击，其中对手盲目地遍历所述令牌/参数的可能值的空间以试图获得优势。攻击者的目标是找到另一个允许的值，以某种方式提升他们在系统中的权限，披露信息或以某种有益于攻击者的方式改变系统的行为。如果应用程序不保护这些关键令牌/参数的完整性，则无法确定这些值是否已被篡改。不应依赖用于保护数据以保密的措施来提供完整性服务。

### Extended Description:

When an application relies on obfuscation or incorrectly applied / weak encryption to protect client-controllable tokens or parameters, that may have an effect on the user state, system state, or some decision made on the server. Without protecting the tokens/parameters for integrity, the application is vulnerable to an attack where an adversary blindly traverses the space of possible values of the said token/parameter in order to attempt to gain an advantage. The goal of the attacker is to find another admissible value that will somehow elevate their privileges in the system, disclose information or change the behavior of the system in some way beneficial to the attacker. If the application does not protect these critical tokens/parameters for integrity, it will not be able to determine that these values have been tampered with. Measures that are used to protect data for confidentiality should not be relied upon to provide the integrity service.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 650

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Trusting HTTP Permission Methods on the Server Side zh: -->信任服务器端的HTTP权限方法*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

服务器包含一种保护机制，该机制假定使用HTTP GET访问的任何URI都不会导致对关联资源的状态更改。这可能允许攻击者绕过预期的访问限制并进行资源修改和删除攻击，因为某些应用程序允许GET修改状态。

### Description:

The server contains a protection mechanism that assumes that any URI that is accessed using HTTP GET will not cause a state change to the associated resource. This might allow attackers to bypass intended access restrictions and conduct resource modification and deletion attacks, since some applications allow GET to modify state.

### 详细描述:

HTTP GET方法和一些其他方法旨在检索资源，而不是改变服务器端的应用程序或资源的状态。此外，HTTP规范要求GET请求（和其他请求）不应有副作用。相信它足以防止意外的资源更改，应用程序可能会禁止HTTP请求对资源表示执行DELETE，PUT和POST操作。但是，HTTP协议本身没有任何内容实际上阻止HTTP GET方法执行的不仅仅是查询数据。开发人员可以轻松编写接受HTTP GET请求的程序，这些程序实际上可以在服务器上创建，更新或删除数据。例如，基于REST的Web服务的常见做法是让HTTP GET请求修改服务器端的资源。但是，只要发生这种情况，就需要在应用程序中正确实施访问控制。不应假设只有HTTP DELETE，PUT，POST和其他方法才有权改变请求中正在访问的资源的表示。

### Extended Description:

The HTTP GET method and some other methods are designed to retrieve resources and not to alter the state of the application or resources on the server side. Furthermore, the HTTP specification requires that GET requests (and other requests) should not have side effects. Believing that it will be enough to prevent unintended resource alterations, an application may disallow the HTTP requests to perform DELETE, PUT and POST operations on the resource representation. However, there is nothing in the HTTP protocol itself that actually prevents the HTTP GET method from performing more than just query of the data. Developers can easily code programs that accept a HTTP GET request that do in fact create, update or delete data on the server. For instance, it is a common practice with REST based Web Services to have HTTP GET requests modifying resources on the server side. However, whenever that happens, the access control needs to be properly enforced in the application. No assumptions should be made that only HTTP DELETE, PUT, POST, and other methods have the power to alter the representation of the resource being accessed in the request.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 651

提交日期 2010-09-27---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Information Exposure Through WSDL File zh: -->信息暴露通过WSDL文件*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web服务体系结构可能需要公开Web服务定义语言（WSDL）文件，该文件包含有关可公开访问的服务的信息以及这些服务的调用者应如何与它们交互（例如，他们期望的参数和返回的类型）。

### Description:

The Web services architecture may require exposing a Web Service Definition Language (WSDL) file that contains information on the publicly accessible services and how callers of these services should interact with them (e.g. what parameters they expect and what types they return).

### 详细描述:

如果符合以下任何条件，则可能会发生信息泄露：  
  
  
WSDL文件可供更广泛的受众访问。  
WSDL文件包含有关不应公开访问的方法/服务的信息或有关已弃用方法的信息。由于WSDL通常是从代码自动生成的，因此更有可能出现此问题。  
WSDL文件中的信息有助于猜测不应公开访问的方法/资源的名称/位置。

### Extended Description:

An information exposure may occur if any of the following apply:  
  
  
The WSDL file is accessible to a wider audience than intended.  
The WSDL file contains information on the methods/services that should not be publicly accessible or information about deprecated methods. This problem is made more likely due to the WSDL often being automatically generated from the code.  
Information in the WSDL file helps guess names/locations of methods/resources that should not be publicly accessible.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 652

提交日期 2008-10-14---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Neutralization of Data within XQuery Expressions ('XQuery Injection') zh: -->XQuery表达式中的数据中和不正确（'XQuery Injection'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用外部输入动态构造用于从XML数据库检索数据的XQuery表达式，但它不会中和或错误地中和该输入。这允许攻击者控制查询的结构。

### Description:

The software uses external input to dynamically construct an XQuery expression used to retrieve data from an XML database, but it does not neutralize or incorrectly neutralizes that input. This allows an attacker to control the structure of the query.

### 详细描述:

实际效果是攻击者可以控制从XML数据库中选择的信息，并可以使用该功能来控制应用程序流，修改逻辑，检索未经授权的数据或绕过重要检查（例如身份验证）。

### Extended Description:

The net effect is that the attacker will have control over the information selected from the XML database and may use that ability to control application flow, modify logic, retrieve unauthorized data, or bypass important checks (e.g. authentication).

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这种弱点类似于启用注入式攻击的其他弱点，例如SQL注入，命令注入和LDAP注入。主要区别在于此处的攻击目标是XML数据库。

Weakness ID: 653

提交日期 2009-01-12---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Insufficient Compartmentalization zh: -->分区不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品没有充分划分需要不同权限级别，权限或权限的功能或进程。

### Description:

The product does not sufficiently compartmentalize functionality or processes that require different privilege levels, rights, or permissions.

### 详细描述:

当较低权限用户可访问的功能出现弱点时，如果没有强大的边界，攻击可能会将损害范围扩大到更高权限的用户。

### Extended Description:

When a weakness occurs in functionality that is accessible by lower-privileged users, then without strong boundaries, an attack might extend the scope of the damage to higher-privileged users.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

与CWE-250（具有不必要特权​​的执行）密切相关。 CWE-653是为每个权限提供单独的组件; CWE-250旨在确保每个组件具有尽可能少的权限。以这种方式，划分成为减少特权的一种机制。

Weakness ID: 654

提交日期 2009-01-12---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Reliance on a Single Factor in a Security Decision zh: -->依赖于安全决策中的单一因素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

保护机制完全或在很大程度上依赖于单个条件或单个对象或实体的完整性的评估，以便做出关于授予对受限资源或功能的访问的决定。

### Description:

A protection mechanism relies exclusively, or to a large extent, on the evaluation of a single condition or the integrity of a single object or entity in order to make a decision about granting access to restricted resources or functionality.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

该节点与“特权分离”一词密切相关。该术语在行业中以几种不同的方式使用，但它们通常结合两个密切相关的原则：区域化（CWE-653）并且在安全决策中仅使用一个因素（该节点）。适当的分区隐式地将多个因素引入到安全决策中，但是可能存在这样的情况：认证或其他不涉及分区的机制需要多个因素，例如对提交的证书执行所有必需的检查。 CWE-653和CWE-654可能会引发进一步的讨论。

Weakness ID: 655

提交日期 2009-01-12---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Insufficient Psychological Acceptability zh: -->心理可接受性不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件具有保护机制，使用起来太困难或不方便，鼓励非恶意用户禁用或绕过该机制，无论是偶然还是故意。

### Description:

The software has a protection mechanism that is too difficult or inconvenient to use, encouraging non-malicious users to disable or bypass the mechanism, whether by accident or on purpose.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这种弱点包括许多安全措施，导致用户不方便，需要努力或造成挫折，这些措施与受保护资产的风险或价值不成比例，或者被认为是无效的。

Weakness ID: 656

提交日期 2009-01-12---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Reliance on Security Through Obscurity zh: -->通过晦涩来依赖安全*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用一种保护机制，其强度在很大程度上取决于其隐蔽性，因此其算法或关键数据的知识足以击败该机制。

### Description:

The software uses a protection mechanism whose strength depends heavily on its obscurity, such that knowledge of its algorithms or key data is sufficient to defeat the mechanism.

### 详细描述:

如果攻击者能够对机制的内部工作进行逆向工程，那么对“通过默默无闻的安全”的依赖会产生由此产生的弱点。请注意，默默无闻可以是防御的一小部分，因为它可以为攻击者创造更多的工作;但是，如果将其作为主要保护手段，则存在重大风险。

### Extended Description:

This reliance on "security through obscurity" can produce resultant weaknesses if an attacker is able to reverse engineer the inner workings of the mechanism. Note that obscurity can be one small part of defense in depth, since it can create more work for an attacker; however, it is a significant risk if used as the primary means of protection.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

请注意，此弱点与CWE-603（使用客户端身份验证）之间存在密切关系。如果开发人员不相信用户可以对客户端进行逆向工程，那么他们更有可能选择客户端身份验证，因为他们认为这是安全的。

Weakness ID: 657

提交日期 I`m don`t know---> 修改日期 2017-01-19

* **Weakness Name:** *en: --> Violation of Secure Design Principles zh: -->违反安全设计原则*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品违反了公认的安全设计原则。

### Description:

The product violates well-established principles for secure design.

### 详细描述:

这可能会导致缺陷或使开发人员更容易在实现过程中引入相关的弱点。因为代码以设计为中心，所以修复设计问题可能是资源密集型的。

### Extended Description:

This can introduce resultant weaknesses or make it easier for developers to introduce related weaknesses during implementation. Because code is centered around design, it can be resource-intensive to fix design problems.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 662

提交日期 2010-09-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Synchronization zh: -->同步不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件尝试以独占方式使用共享资源，但不会阻止或错误地阻止另一个线程或进程使用该资源。

### Description:

The software attempts to use a shared resource in an exclusive manner, but does not prevent or incorrectly prevents use of the resource by another thread or process.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 663

提交日期 2010-09-27---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of a Non-reentrant Function in a Concurrent Context zh: -->在并发上下文中使用非重入函数*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在并发上下文中调用非重入函数，其中竞争代码序列（例如，线程或信号处理程序）可能有机会调用相同的函数或以其他方式影响其状态。

### Description:

The software calls a non-reentrant function in a concurrent context in which a competing code sequence (e.g. thread or signal handler) may have an opportunity to call the same function or otherwise influence its state.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 664

提交日期 2009-05-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Control of a Resource Through its Lifetime zh: -->资源的生命周期控制不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在整个创建，使用和发布的生命周期内，该软件不会维护或错误地保持对资源的控制。

### Description:

The software does not maintain or incorrectly maintains control over a resource throughout its lifetime of creation, use, and release.

### 详细描述:

资源通常有关于如何创建，使用和销毁的明确说明。当软件不遵循这些说明时，可能会导致意外行为和潜在的可利用状态。  
即使没有明确的指示，也应该遵守各种原则，例如“在创建完成之前不要使用对象”，或“在对象被破坏后不使用对象”。

### Extended Description:

Resources often have explicit instructions on how to be created, used and destroyed. When software does not follow these instructions, it can lead to unexpected behaviors and potentially exploitable states.  
Even without explicit instructions, various principles are expected to be adhered to, such as "Do not use an object until after its creation is complete," or "do not use an object after it has been slated for destruction."

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此节点及其子节点需要做更多的工作。存在观点/分层问题;例如，一个细分基于生命周期阶段（CWE-404，CWE-665），而其他儿童则独立于生命周期，例如CWE-400。其他人没有指定尽可能多的基础或变体，例如CWE-704，其主要涵盖此阶段的数字。

Weakness ID: 665

提交日期 2009-01-12---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Initialization zh: -->初始化不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会初始化或错误地初始化资源，这可能会在访问或使用资源时使资源处于意外状态。

### Description:

The software does not initialize or incorrectly initializes a resource, which might leave the resource in an unexpected state when it is accessed or used.

### 详细描述:

当关联资源预期具有某些属性或值（例如确定用户是否已经过身份验证的变量）时，这会产生安全隐患。

### Extended Description:

This can have security implications when the associated resource is expected to have certain properties or values, such as a variable that determines whether a user has been authenticated or not.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 666

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Operation on Resource in Wrong Phase of Lifetime zh: -->终身错误阶段的资源运作*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在资源生命周期的错误阶段对资源执行操作，这可能导致意外行为。

### Description:

The software performs an operation on a resource at the wrong phase of the resource's lifecycle, which can lead to unexpected behaviors.

### 详细描述:

当开发人员想要初始化，使用或释放​​资源时，遵循有关如何操作该资源的规范并确保资源处于预期状态非常重要。在这种情况下，软件希望在资源处于其生命周期的错误阶段时对资源执行通常有效的操作，初始化，使用或释放​​。

### Extended Description:

When a developer wants to initialize, use or release a resource, it is important to follow the specifications outlined for how to operate on that resource and to ensure that the resource is in the expected state. In this case, the software wants to perform a normally valid operation, initialization, use or release, on a resource when it is in the incorrect phase of its lifetime.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 667

提交日期 2010-12-13---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Locking zh: -->锁定不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法正确获取资源锁定，或者未正确释放资源上的锁定，从而导致意外的资源状态更改和行为。

### Description:

The software does not properly acquire a lock on a resource, or it does not properly release a lock on a resource, leading to unexpected resource state changes and behaviors.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 668

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Exposure of Resource to Wrong Sphere zh: -->资源暴露于错误的领域*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品将资源暴露给错误的控制范围，从而为非预期的参与者提供对资源的不适当访问。

### Description:

The product exposes a resource to the wrong control sphere, providing unintended actors with inappropriate access to the resource.

### 详细描述:

诸如文件和目录之类的资源可能会通过诸如不安全的权限之类的机制无意中暴露，或者当程序意外地对错误的对象进行操作时。例如，程序可能希望私有文件只能提供给特定用户。这有效地定义了一个控制范围，旨在防止攻击者访问这些私有文件。如果文件权限不安全，则除用户之外的其他方将能够访问这些文件。  
单独的控制范围可能有效地要求用户只能访问私有文件，而不能访问系统上的任何其他文件。如果程序不确保用户仅请求私有文件，则用户可能能够访问系统上的其他文件。  
在任何一种情况下，最终结果都是资源暴露给了错误的一方。

### Extended Description:

Resources such as files and directories may be inadvertently exposed through mechanisms such as insecure permissions, or when a program accidentally operates on the wrong object. For example, a program may intend that private files can only be provided to a specific user. This effectively defines a control sphere that is intended to prevent attackers from accessing these private files. If the file permissions are insecure, then parties other than the user will be able to access those files.  
A separate control sphere might effectively require that the user can only access the private files, but not any other files on the system. If the program does not ensure that the user is only requesting private files, then the user might be able to access other files on the system.  
In either case, the end result is that a resource has been exposed to the wrong party.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

“控制范围”是一组资源和行为，可供单个参与者或一组参与者访问。产品的安全模型通常会隐式定义多个领域。例如，服务器可以为“管理员”定义一个球体，该管理员可以在/ home / server /下创建具有子目录的新用户帐户，第二个球体可以覆盖可以在其自己的子目录中创建或删除文件的用户集。第三个范围可能是“对安装产品的操作系统进行身份验证的用户”。每个球体都有不同的演员和允许的行为。

Weakness ID: 669

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incorrect Resource Transfer Between Spheres zh: -->球体之间的资源转移不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品无法正确地将资源/行为转移到另一个领域，或者以不当方式控制该资源的方式从另一个领域不正确地导入资源/行为。

### Description:

The product does not properly transfer a resource/behavior to another sphere, or improperly imports a resource/behavior from another sphere, in a manner that provides unintended control over that resource.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

“控制范围”是一组资源和行为，可供单个参与者或一组参与者访问。产品的安全模型通常会隐式定义多个领域。例如，服务器可以为“管理员”定义一个球体，该管理员可以在/ home / server /下创建具有子目录的新用户帐户，第二个球体可以覆盖可以在其自己的子目录中创建或删除文件的用户集。第三个范围可能是“对安装产品的操作系统进行身份验证的用户”。每个球体都有不同的演员和允许的行为。

### 笔记 (Notes):

没有笔记

Weakness ID: 670

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Always-Incorrect Control Flow Implementation zh: -->始终不正确的控制流实施*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码包含一个控制流路径，该路径不反映路径要实现的算法，导致在导航此路径时出现错误行为。

### Description:

The code contains a control flow path that does not reflect the algorithm that the path is intended to implement, leading to incorrect behavior any time this path is navigated.

### 详细描述:

这个弱点捕获了特定代码段相对于它正在实现的算法总是不正确的情况。例如，如果C程序员打算在单个块中包含多个语句但不包括括号（CWE-483），那么逻辑总是不正确的。此问题与代码通常正常运行的大多数弱点形成对比，除非它以恶意方式进行外部操作。

### Extended Description:

This weakness captures cases in which a particular code segment is always incorrect with respect to the algorithm that it is implementing. For example, if a C programmer intends to include multiple statements in a single block but does not include the enclosing braces (CWE-483), then the logic is always incorrect. This issue is in contrast to most weaknesses in which the code usually behaves correctly, except when it is externally manipulated in malicious ways.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此节点可能会拆分为较低级别的节点。 “早期返回”用于过早地将控制权返回给呼叫者（例如，CWE-584）。 “超额返回”是指控制在调用堆栈中返回太远（CWE-600，CWE-395）。当产品将控制保持在较低的执行级别时，控制应该“进一步”返回到调用堆栈（CWE-455）时，会发生“控制不正确”。 “不正确的语法”涵盖了“完全错误”的代码，例如CWE-484和CWE-483。

Weakness ID: 671

提交日期 2009-01-12---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Lack of Administrator Control over Security zh: -->缺乏管理员对安全性的控制*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用安全功能，以防止产品管理员定制安全设置以反映产品使用的环境。这会导致产生的弱点或阻止其在管理员所需的安全级别上运行。

### Description:

The product uses security features in a way that prevents the product's administrator from tailoring security settings to reflect the environment in which the product is being used. This introduces resultant weaknesses or prevents it from operating at a level of security that is desired by the administrator.

### 详细描述:

如果产品的管理员无法始终管理与安全相关的决策，则可能无法保护产品免受外部威胁（包括产品开发人员）的影响。例如，管理员无法更改硬编码的帐户名和密码，从而使该产品暴露给管理员无法阻止的攻击。

### Extended Description:

If the product's administrator does not have the ability to manage security-related decisions at all times, then protecting the product from outside threats - including the product's developer - can become impossible. For example, a hard-coded account name and password cannot be changed by the administrator, thus exposing that product to attacks that the administrator can not prevent.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 672

提交日期 2010-02-16---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Operation on a Resource after Expiration or Release zh: -->到期或释放后对资源的操作*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在该资源过期，释放或撤销之后，该软件使用，访问或以其他方式操作资源。

### Description:

The software uses, accesses, or otherwise operates on a resource after that resource has been expired, released, or revoked.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 673

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> External Influence of Sphere Definition zh: -->球体定义的外部影响*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品不会阻止外部参与者对控制领域的定义。

### Description:

The product does not prevent the definition of control spheres from external actors.

### 详细描述:

通常，产品在代码本身内定义其控制范围，或通过产品管理员的配置定义。在某些情况下，外部方可以更改控制范围的定义。这通常是由此产生的弱点。

### Extended Description:

Typically, a product defines its control sphere within the code itself, or through configuration by the product's administrator. In some cases, an external party can change the definition of the control sphere. This is typically a resultant weakness.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

“控制范围”是一组资源和行为，可供单个参与者或一组参与者访问。产品的安全模型通常会隐式定义多个领域。例如，服务器可以为“管理员”定义一个球体，该管理员可以在/ home / server /下创建具有子目录的新用户帐户，第二个球体可以覆盖可以在其自己的子目录中创建或删除文件的用户集。第三个范围可能是“对安装产品的操作系统进行身份验证的用户”。每个球体都有不同的演员和允许的行为。

Weakness ID: 674

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Uncontrolled Recursion zh: -->不受控制的递归*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品无法正确控制发生的递归量，这会消耗过多的资源，例如分配的内存或程序堆栈。

### Description:

The product does not properly control the amount of recursion that takes place, which consumes excessive resources, such as allocated memory or the program stack.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 675

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Duplicate Operations on Resource zh: -->资源重复操作*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当操作仅应用一次时，产品对资源执行相同的操作两次或更多次。

### Description:

The product performs the same operation on a resource two or more times, when the operation should only be applied once.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这个弱点可能与其他与倍增相关的问题密切相关，例如CWE-462（alist中的重复键）或CWE-102（Struts重复验证表单）。这通常是违反API合同的情况（CWE-227）。

Weakness ID: 676

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Potentially Dangerous Function zh: -->使用潜在危险函数*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序调用一个潜在的危险函数，如果使用不当可能会引入漏洞，但该函数也可以安全使用。

### Description:

The program invokes a potentially dangerous function that could introduce a vulnerability if it is used incorrectly, but the function can also be used safely.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这个弱点与CWE-242（使用固有危险功能）不同。 CWE-242涵盖了具有如此重大安全问题的功能，以至于无法保证它们是安全的。如果使用得当，某些功能不会直接带来安全风险，但如果未正确调用则会引入弱点。这些被认为是有潜在危险的。一个众所周知的例子是strcpy（）函数。当提供的目标缓冲区大于其源时，strcpy（）不会溢出。但是，一些开发人员完全禁止strcpy（），因此经常被滥用。

Weakness ID: 680

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Integer Overflow to Buffer Overflow zh: -->整数溢出到缓冲区溢出*
* **Abstraction:** *Compound* **Structure:** *Chain* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品执行计算以确定要分配多少内存，但可能发生整数溢出，导致分配的内存少于预期，从而导致缓冲区溢出。

### Description:

The product performs a calculation to determine how much memory to allocate, but an integer overflow can occur that causes less memory to be allocated than expected, leading to a buffer overflow.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 681

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Conversion between Numeric Types zh: -->数字类型之间的转换不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

从一种数据类型转换为另一种数据类型（例如long到整数）时，可以以产生意外值的方式省略或转换数据。如果在敏感上下文中使用结果值，则可能会发生危险行为。

### Description:

When converting from one data type to another, such as long to integer, data can be omitted or translated in a way that produces unexpected values. If the resulting values are used in a sensitive context, then dangerous behaviors may occur.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 682

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Calculation zh: -->计算不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件执行的计算会生成不正确或意外的结果，这些结果稍后将用于安全关键决策或资源管理。

### Description:

The software performs a calculation that generates incorrect or unintended results that are later used in security-critical decisions or resource management.

### 详细描述:

当软件错误地执行安全关键计算时，可能会导致不正确的资源分配，不正确的权限分配或其他事项之间的比较失败。计算错误的许多直接结果可能导致更大的问题，例如保护机制失败甚至任意代码执行。

### Extended Description:

When software performs a security-critical calculation incorrectly, it might lead to incorrect resource allocations, incorrect privilege assignments, or failed comparisons among other things. Many of the direct results of an incorrect calculation can lead to even larger problems such as failed protection mechanisms or even arbitrary code execution.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 683

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Function Call With Incorrect Order of Arguments zh: -->函数调用具有不正确的参数顺序*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件调用函数，过程或例程，但调用者以不正确的顺序指定参数，从而导致产生的弱点。

### Description:

The software calls a function, procedure, or routine, but the caller specifies the arguments in an incorrect order, leading to resultant weaknesses.

### 详细描述:

虽然编译器可能会在某些语言中捕获此弱点，但在被调用函数接受可变数量或类型的参数（例如C中的格式字符串）的情况下，它可能会更频繁地发生。它也可能出现在语言或环境中不强制强打字。

### Extended Description:

While this weakness might be caught by the compiler in some languages, it can occur more frequently in cases in which the called function accepts variable numbers or types of arguments, such as format strings in C. It also can occur in languages or environments that do not enforce strong typing.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 684

提交日期 2011-03-29---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Provision of Specified Functionality zh: -->提供的指定功能不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码根据其发布的规范不起作用，可能导致错误使用。

### Description:

The code does not function according to its published specifications, potentially leading to incorrect usage.

### 详细描述:

向外部方提供功能时，软件的行为必须符合指定的详细信息。如果没有记录细微差别的要求，则该功能可能会对调用者产生意外行为，从而可能导致可利用状态。

### Extended Description:

When providing functionality to an external party, it is important that the software behaves in accordance with the details specified. When requirements of nuances are not documented, the functionality may produce unintended behaviors for the caller, possibly leading to an exploitable state.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 685

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Function Call With Incorrect Number of Arguments zh: -->函数调用参数数量不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件调用函数，过程或例程，但调用者指定的参数太多或参数太少，这可能导致未定义的行为和由此产生的弱点。

### Description:

The software calls a function, procedure, or routine, but the caller specifies too many arguments, or too few arguments, which may lead to undefined behavior and resultant weaknesses.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 686

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Function Call With Incorrect Argument Type zh: -->函数调用具有不正确的参数类型*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件调用函数，过程或例程，但调用者指定的参数是错误的数据类型，这可能导致产生的弱点。

### Description:

The software calls a function, procedure, or routine, but the caller specifies an argument that is the wrong data type, which may lead to resultant weaknesses.

### 详细描述:

这种弱点最有可能发生在松散类型的语言中，或者在强类型语言中，其中在编译时不能强制执行变量参数的类型，或者存在隐式转换。

### Extended Description:

This weakness is most likely to occur in loosely typed languages, or in strongly typed languages in which the types of variable arguments cannot be enforced at compilation time, or where there is implicit casting.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 687

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Function Call With Incorrectly Specified Argument Value zh: -->函数调用具有错误指定的参数值*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件调用函数，过程或例程，但调用者指定包含错误值的参数，这可能导致产生的弱点。

### Description:

The software calls a function, procedure, or routine, but the caller specifies an argument that contains the wrong value, which may lead to resultant weaknesses.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

初级时，这种弱点最有可能发生在很少测试的代码中，因为错误的值可能会改变程序执行的语义，导致明显不正确的行为。它也可能是程序将错误的值赋给变量的问题的结果，并且该变量稍后在函数调用中使用。从这个意义上说，这个问题可以被认为是与CWE中的许多实现错误具有链接关系。

Weakness ID: 688

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Function Call With Incorrect Variable or Reference as Argument zh: -->函数调用具有不正确的变量或引用作为参数*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件调用函数，过程或例程，但调用者将错误的变量或引用指定为参数之一，这可能导致未定义的行为和由此产生的弱点。

### Description:

The software calls a function, procedure, or routine, but the caller specifies the wrong variable or reference as one of the arguments, which may lead to undefined behavior and resultant weaknesses.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 689

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Permission Race Condition During Resource Copy zh: -->资源复制期间的权限竞争条件*
* **Abstraction:** *Compound* **Structure:** *Composite* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在复制或克隆资源时，产品在复制完成之前不会设置资源的权限或访问控制，从而在复制过程中将资源暴露给其他领域。

### Description:

The product, while copying or cloning a resource, does not set the resource's permissions or access control until the copy is complete, leaving the resource exposed to other spheres while the copy is taking place.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在深入研究的。这种弱点似乎可能发生在复杂或大型复制操作发生的任何情况下，当资源可以在创建后立即可用于其他领域，但在其初始化完成之前。

Weakness ID: 690

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Unchecked Return Value to NULL Pointer Dereference zh: -->未选中的返回值为NULL指针取消引用*
* **Abstraction:** *Compound* **Structure:** *Chain* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

如果函数失败，则在调用可以使用NULL指针返回的函数后，产品不会检查错误，这会导致生成的NULL指针取消引用。

### Description:

The product does not check for an error after calling a function that can return with a NULL pointer if the function fails, which leads to a resultant NULL pointer dereference.

### 详细描述:

虽然未经检查的返回值弱点不仅限于NULL指针的返回（参见CWE-252中的示例），但函数通常返回NULL以指示错误状态。如果未选中此错误条件，则可能会发生NULL指针取消引用。

### Extended Description:

While unchecked return value weaknesses are not limited to returns of NULL pointers (see the examples in CWE-252), functions often return NULL to indicate an error status. When this error condition is not checked, a NULL pointer dereference can occur.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 691

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Insufficient Control Flow Management zh: -->控制流量管理不足*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码在执行期间不能充分管理其控制流，从而创建可以以意外方式修改控制流的条件。

### Description:

The code does not sufficiently manage its control flow during execution, creating conditions in which the control flow can be modified in unexpected ways.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这是一个相当高级的概念，虽然它涵盖了CWE中的许多弱点，这些弱点在草案9发布之前在整个研究视图（CWE-1000）中更加分散。

Weakness ID: 692

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incomplete Blacklist to Cross-Site Scripting zh: -->不完整的黑名单到跨站点脚本*
* **Abstraction:** *Compound* **Structure:** *Chain* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用基于黑名单的保护机制来抵御XSS攻击，但黑名单不完整，允许XSS变种成功。

### Description:

The product uses a blacklist-based protection mechanism to defend against XSS attacks, but the blacklist is incomplete, allowing XSS variants to succeed.

### 详细描述:

虽然XSS可能看起来很容易预防，但Web浏览器解析网页的方式差异很大，黑名单无法跟踪所有变化。 “XSS备忘单”[REF-564]包含大量旨在绕过不完整黑名单的攻击。

### Extended Description:

While XSS might seem simple to prevent, web browsers vary so widely in how they parse web pages, that a blacklist cannot keep track of all the variations. The "XSS Cheat Sheet" [REF-564] contains a large number of attacks that are intended to bypass incomplete blacklists.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 693

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Protection Mechanism Failure zh: -->保护机制失败*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品不使用或错误地使用保护机制，以提供足够的防御以防止对产品的定向攻击。

### Description:

The product does not use or incorrectly uses a protection mechanism that provides sufficient defense against directed attacks against the product.

### 详细描述:

这种弱点包括三种不同的情况。当应用程序没有定义针对某类攻击的任何机制时，会发生“丢失”保护机制。 “不充分”的保护机制可能会提供一些防御 - 例如，针对最常见的攻击 - 但它并不能防止所有预期的攻击。最后，当一个机制可用并在产品中主动使用时，会发生“忽略”机制，但开发人员尚未在某些代码路径中应用它。

### Extended Description:

This weakness covers three distinct situations. A "missing" protection mechanism occurs when the application does not define any mechanism against a certain class of attack. An "insufficient" protection mechanism might provide some defenses - for example, against the most common attacks - but it does not protect against everything that is intended. Finally, an "ignored" mechanism occurs when a mechanism is available and in active use within the product, but the developer has not applied it in some code path.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这是一个相当高级的概念，尽管它涵盖了CWE中的一些弱点，这些弱点在草案9发布之前更加分散在整个自然层级中。

Weakness ID: 694

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of Multiple Resources with Duplicate Identifier zh: -->使用具有重复标识符的多个资源*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在需要唯一标识符的上下文中使用可具有相同标识符的多个资源。

### Description:

The software uses multiple resources that can have the same identifier, in a context in which unique identifiers are required.

### 详细描述:

如果软件假定每个资源都具有唯一标识符，则如果攻击者可以使多个资源与同一标识符相关联，则软件可以在错误的资源上运行。

### Extended Description:

If the software assumes that each resource has a unique identifier, the software could operate on the wrong resource if attackers can cause multiple resources to be associated with the same identifier.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这种弱点可能与其他与倍增相关的问题密切相关，例如CWE-675（资源重复操作）。这通常是违反API合同的情况（CWE-227）。

Weakness ID: 695

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of Low-Level Functionality zh: -->使用低级功能*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用软件应该运行的框架或规范明确禁止的低级功能。

### Description:

The software uses low-level functionality that is explicitly prohibited by the framework or specification under which the software is supposed to operate.

### 详细描述:

使用低级功能可能会以意外的方式违反规范，从而有效地禁用内置保护机制，引入可利用的不一致性或以其他方式暴露攻击功能。

### Extended Description:

The use of low-level functionality can violate the specification in unexpected ways that effectively disable built-in protection mechanisms, introduce exploitable inconsistencies, or otherwise expose the functionality to attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 696

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Behavior Order zh: -->行为顺序不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件执行多个相关行为，但行为以错误的顺序执行，可能产生结果的弱点。

### Description:

The software performs multiple related behaviors, but the behaviors are performed in the wrong order in ways which may produce resultant weaknesses.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 697

提交日期 2018-03-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Comparison zh: -->比较不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在安全相关的上下文中比较两个实体，但比较不正确，这可能导致产生的弱点。

### Description:

The software compares two entities in a security-relevant context, but the comparison is incorrect, which may lead to resultant weaknesses.

### 详细描述:

这个弱点课程涵盖了几种可能性：  
  
  
比较检查一个因素不正确;  
比较应考虑多种因素，但根本不检查其中的一些因素;  
比较检查错误的因素。

### Extended Description:

This weakness class covers several possibilities:  
  
  
the comparison checks one factor incorrectly;  
the comparison should consider multiple factors, but it does not check some of those factors at all;  
the comparison checks the wrong factor.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目可能与区分大小写（CWE-178）有一些关系，但除了比较之外，区分大小写是其他类型弱点的一个因素。此外，在密码学中，当某些比较操作不是在恒定时间内发生时，某些攻击是可能的，导致与时间相关的信息泄漏（CWE-208）。

Weakness ID: 698

提交日期 2013-02-21---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Execution After Redirect (EAR) zh: -->重定向后执行（EAR）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序将重定向发送到另一个位置，但它不是退出，而是执行其他代码。

### Description:

The web application sends a redirect to another location, but instead of exiting, it executes additional code.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 703

提交日期 2010-12-13---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Check or Handling of Exceptional Conditions zh: -->不正确的检查或处理特殊情况*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法正确预测或处理在软件正常运行期间很少发生的异常情况。

### Description:

The software does not properly anticipate or handle exceptional conditions that rarely occur during normal operation of the software.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这是一个高级类，可能与其他类有一些重叠。可以说，即使是“正常”的弱点，如缓冲区溢出，也会出现异常或异常情况。从这个意义上讲，这可能是CWE中大多数其他弱点的固有方面，类似于API滥用（CWE-227）和不良代码质量指标（CWE-398）。但是，此条目目前旨在统一研究概念视图（CWE-1000）中没有其他位置的不同概念。

Weakness ID: 704

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Type Conversion or Cast zh: -->类型转换或转换不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法将对象，资源或结构从一种类型正确转换为另一种类型。

### Description:

The software does not correctly convert an object, resource, or structure from one type to a different type.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 705

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Control Flow Scoping zh: -->控制流程范围不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在完成任务或检测到异常情况后，软件无法将控制流正确地返回到正确的位置。

### Description:

The software does not properly return control flow to the proper location after it has completed a task or detected an unusual condition.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 706

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Incorrectly-Resolved Name or Reference zh: -->使用错误解析的名称或参考*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用名称或引用来访问资源，但名称/引用解析为预期控制范围之外的资源。

### Description:

The software uses a name or reference to access a resource, but the name/reference resolves to a resource that is outside of the intended control sphere.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 707

提交日期 2009-05-27---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Enforcement of Message or Data Structure zh: -->消息或数据结构的不正确执行*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在从上游组件读取或发送到下游组件之前，软件不强制执行或错误地强制执行结构化消息或数据格式良好。

### Description:

The software does not enforce or incorrectly enforces that structured messages or data are well-formed before being read from an upstream component or sent to a downstream component.

### 详细描述:

如果消息格式错误，则可能导致错误地解释消息。  
这种弱点通常适用于产品准备另一个进程必须采取行动的控制消息（例如命令或查询）以及旨在作为数据的恶意输入可以进入控制平面的情况。然而，这种弱点也适用于并非总是存在控制影响的更一般情况。

### Extended Description:

If a message is malformed it may cause the message to be incorrectly interpreted.  
This weakness typically applies in cases where the product prepares a control message that another process must act on, such as a command or query, and malicious input that was intended as data, can enter the control plane instead. However, this weakness also applies to more general cases where there are not always control implications.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 708

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incorrect Ownership Assignment zh: -->所有权分配不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件将所有者分配给资源，但所有者不在预期的控制范围内。

### Description:

The software assigns an owner to a resource, but the owner is outside of the intended control sphere.

### 详细描述:

这可以允许资源由预期控制范围之外的参与者操纵。

### Extended Description:

This may allow the resource to be manipulated by actors outside of the intended control sphere.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这会重叠验证错误，权限和权限。一个密切相关的弱点是不正确地将组分配给资源。目前尚不清楚它是否属于此条目或需要不同的条目。

Weakness ID: 710

提交日期 2017-11-08---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Adherence to Coding Standards zh: -->对编码标准的不当遵守*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不遵循某些开发编码规则，这可能导致由此产生的弱点或增加相关漏洞的严重性。

### Description:

The software does not follow certain coding rules for development, which can lead to resultant weaknesses or increase the severity of the associated vulnerabilities.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 732

提交日期 2009-01-12---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Permission Assignment for Critical Resource zh: -->关键资源的权限分配不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件以允许非预期的actor读取或修改该资源的方式指定安全关键资源的权限。

### Description:

The software specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.

### 详细描述:

当为资源提供权限设置以提供对超出要求的更广泛的参与者的访问时，它可能导致敏感信息的暴露，或者由非预期方修改该资源。当资源与程序配置，执行或敏感用户数据相关时，这尤其危险。

### Extended Description:

When a resource is given a permissions setting that provides access to a wider range of actors than required, it could lead to the exposure of sensitive information, or the modification of that resource by unintended parties. This is especially dangerous when the resource is related to program configuration, execution or sensitive user data.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

特权，权限和参与者（例如用户和组）之间的关系需要在Research视图中进一步细化。一个复杂因素是这些概念适用于两个不同的支柱，涉及资源控制（CWE-664）和保护机制故障（CWE-396）。

Weakness ID: 733

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Compiler Optimization Removal or Modification of Security-critical Code zh: -->编译器优化删除或修改安全关键代码*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

开发人员在软件中构建了一个安全关键的保护机制，但编译器优化了程序，以便删除或修改该机制。

### Description:

The developer builds a security-critical protection mechanism into the software, but the compiler optimizes the program such that the mechanism is removed or modified.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 749

提交日期 2009-01-12---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Exposed Dangerous Method or Function zh: -->暴露的危险方法或功能*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件提供了一个应用程序编程接口（API）或类似的接口，用于与外部参与者进行交互，但该接口包含一个未被适当限制的危险方法或功能。

### Description:

The software provides an Applications Programming Interface (API) or similar interface for interaction with external actors, but the interface includes a dangerous method or function that is not properly restricted.

### 详细描述:

根据暴露方法的行为，这种弱点可能导致各种各样的弱点。它可以应用于任何数量的技术和方法，例如ActiveX控件，Java函数，IOCTL等。  
暴露可以通过几种不同的方式发生：  
  
1）功能/方法从未打算暴露给外部参与者。  
2）功能/方法仅供有限的一组参与者访问，例如来自单个网站的基于因特网的访问。

### Extended Description:

This weakness can lead to a wide variety of resultant weaknesses, depending on the behavior of the exposed method. It can apply to any number of technologies and approaches, such as ActiveX controls, Java functions, IOCTLs, and so on.  
The exposure can occur in a few different ways:  
  
1) The function/method was never intended to be exposed to outside actors.  
2) The function/method was only intended to be accessible to a limited set of actors, such as Internet-based access from a single web site.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

报告不足和研究不足。这种弱点可能出现在允许程序员为外部各方提供功能接口的任何技术，语言或框架中，但它没有大量报道。 2007年，CVE开始显示ActiveX应用程序中暴露的方法漏洞报告以及IOCTL访问操作系统级资源的显着增加。已经在各种安全编程源中记录了Java应用程序的这些弱点，但CVE中的报告很少，这表明漏洞研究社区的大多数部分都存在有限的意识。

Weakness ID: 754

提交日期 2010-02-16---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Check for Unusual or Exceptional Conditions zh: -->不正确检查异常或特殊情况*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件不会检查或不正确地检查在软件的日常操作期间不会经常发生的异常或异常情况。

### Description:

The software does not check or improperly checks for unusual or exceptional conditions that are not expected to occur frequently during day to day operation of the software.

### 详细描述:

程序员可以假设某些事件或条件永远不会发生或者不需要担心，例如内存条件低，由于限制性权限而无法访问资源，或者行为不当的客户端或组件。但是，攻击者可能会故意触发这些异常情况，从而违反程序员的假设，可能会引入不稳定，不正确的行为或漏洞。  
请注意，此条目并非仅限于使用异常和异常处理，这些异常和异常处理是检查和处理异常或意外情况的机制。

### Extended Description:

The programmer may assume that certain events or conditions will never occur or do not need to be worried about, such as low memory conditions, lack of access to resources due to restrictive permissions, or misbehaving clients or components. However, attackers may intentionally trigger these unusual conditions, thus violating the programmer's assumptions, possibly introducing instability, incorrect behavior, or a vulnerability.  
Note that this entry is not exclusively about the use of exceptions and exception handling, which are mechanisms for both checking and handling unusual or unexpected conditions.

### 问题背景 (Background Detail):

许多功能将为其行动的成功返回一些价值。这将提醒程序是否处理由该功能引起的任何错误。

### 笔记 (Notes):

有时，当返回值可用于指示错误时，未经检查的返回值是缺少应用程序层检查的特殊条件的代码层实例。但是，并不总是需要返回值来传达异常情况。例如，资源的到期，通过引用传递的值，异步修改的数据，套接字等可以指示不使用返回值的异常条件。

Weakness ID: 755

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Handling of Exceptional Conditions zh: -->对特殊情况的处理不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件无法处理或错误处理异常情况。

### Description:

The software does not handle or incorrectly handles an exceptional condition.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 756

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Missing Custom Error Page zh: -->缺少自定义错误页面*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会向用户返回自定义错误页面，可能会泄露敏感信息。

### Description:

The software does not return custom error pages to the user, possibly exposing sensitive information.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 757

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Selection of Less-Secure Algorithm During Negotiation ('Algorithm Downgrade') zh: -->在协商过程中选择安全性较低的算法（'算法降级'）*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

协议或其实现支持多个参与者之间的交互，并允许这些参与者协商哪个算法应该用作保护机制，例如加密或认证，但是它不选择双方都可用的最强算法。

### Description:

A protocol or its implementation supports interaction between multiple actors and allows those actors to negotiate which algorithm should be used as a protection mechanism such as encryption or authentication, but it does not select the strongest algorithm that is available to both parties.

### 详细描述:

当可以强制安全机制降级以使用安全性较低的算法时，这可以使攻击者更容易通过利用较弱的算法来破坏软件。受害者可能不知道正在使用安全性较低的算法。例如，如果攻击者可以强制通信通道使用明文而不是强加密数据，则攻击者可以通过嗅探来读取通道，而不是通过使用强力技术尝试解密数据的额外努力。

### Extended Description:

When a security mechanism can be forced to downgrade to use a less secure algorithm, this can make it easier for attackers to compromise the software by exploiting weaker algorithm. The victim might not be aware that the less secure algorithm is being used. For example, if an attacker can force a communications channel to use cleartext instead of strongly-encrypted data, then the attacker could read the channel by sniffing, instead of going through extra effort of trying to decrypt the data using brute force techniques.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这与CWE-300（中间人）有关，虽然并非所有降级攻击都必然需要中间人。见例子。

Weakness ID: 758

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Reliance on Undefined, Unspecified, or Implementation-Defined Behavior zh: -->依赖未定义，未指定或实现定义的行为*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件以某种方式使用API​​函数，数据结构或其他实体，这种方式依赖于并不总是保证为该实体保留的属性。

### Description:

The software uses an API function, data structure, or other entity in a way that relies on properties that are not always guaranteed to hold for that entity.

### 详细描述:

当所需的属性发生变化时，例如当软件移植到不同的平台或发生交互错误（CWE-435）时，这可能导致产生的弱点。

### Extended Description:

This can lead to resultant weaknesses when the required properties change, such as when the software is ported to a different platform or if an interaction error (CWE-435) occurs.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 759

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Use of a One-Way Hash without a Salt zh: -->使用没有盐的单向哈希*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件对不应该是可逆的输入使用单向加密哈希，例如密码，但软件也不使用salt作为输入的一部分。

### Description:

The software uses a one-way cryptographic hash against an input that should not be reversible, such as a password, but the software does not also use a salt as part of the input.

### 详细描述:

这使得攻击者可以更容易地使用字典攻击技术（如彩虹表）预先计算哈希值。  
应该注意的是，尽管普遍认为，使用带有哈希的好盐并不足以增加针对个人密码的攻击者的努力，或者具有大量可用计算资源的攻击者，例如云基于服务或专业，廉价的硬件。如果散列函数的计算成本不高，则脱机密码破解仍然有效;许多加密函数被设计为高效且易受使用大量计算资源的攻击，即使哈希密码强大也是如此。与其他策略（如自适应散列函数）相比，使用salt只会略微增加攻击者的计算要求。有关详细信息，请参阅CWE-916。

### Extended Description:

This makes it easier for attackers to pre-compute the hash value using dictionary attack techniques such as rainbow tables.  
It should be noted that, despite common perceptions, the use of a good salt with a hash does not sufficiently increase the effort for an attacker who is targeting an individual password, or who has a large amount of computing resources available, such as with cloud-based services or specialized, inexpensive hardware. Offline password cracking can still be effective if the hash function is not expensive to compute; many cryptographic functions are designed to be efficient and can be vulnerable to attacks using massive computing resources, even if the hash is cryptographically strong. The use of a salt only slightly increases the computing requirements for an attacker compared to other strategies such as adaptive hash functions. See CWE-916 for more details.

### 问题背景 (Background Detail):

在密码学中，salt指的是在散列之前将一些数据随机添加到输入以使字典攻击更加困难。

### 笔记 (Notes):

没有笔记

Weakness ID: 760

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Use of a One-Way Hash with a Predictable Salt zh: -->使用具有可预测盐的单向哈希*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件对不应该是可逆的输入使用单向加密哈希，例如密码，但软件使用可预测的盐作为输入的一部分。

### Description:

The software uses a one-way cryptographic hash against an input that should not be reversible, such as a password, but the software uses a predictable salt as part of the input.

### 详细描述:

这使得攻击者可以更容易地使用诸如彩虹表之类的字典攻击技术预先计算哈希值，从而有效地禁用不可预测的盐将提供的保护。  
应该注意的是，尽管普遍认为，使用带有哈希的好盐并不足以增加针对个人密码的攻击者的努力，或者具有大量可用计算资源的攻击者，例如云基于服务或专业，廉价的硬件。如果散列函数的计算成本不高，则脱机密码破解仍然有效;许多加密函数被设计为高效且易受使用大量计算资源的攻击，即使哈希密码强大也是如此。与其他策略（如自适应散列函数）相比，使用salt只会略微增加攻击者的计算要求。有关详细信息，请参阅CWE-916。

### Extended Description:

This makes it easier for attackers to pre-compute the hash value using dictionary attack techniques such as rainbow tables, effectively disabling the protection that an unpredictable salt would provide.  
It should be noted that, despite common perceptions, the use of a good salt with a hash does not sufficiently increase the effort for an attacker who is targeting an individual password, or who has a large amount of computing resources available, such as with cloud-based services or specialized, inexpensive hardware. Offline password cracking can still be effective if the hash function is not expensive to compute; many cryptographic functions are designed to be efficient and can be vulnerable to attacks using massive computing resources, even if the hash is cryptographically strong. The use of a salt only slightly increases the computing requirements for an attacker compared to other strategies such as adaptive hash functions. See CWE-916 for more details.

### 问题背景 (Background Detail):

在密码学中，salt指的是在散列之前将一些数据随机添加到输入以使字典攻击更加困难。

### 笔记 (Notes):

没有笔记

Weakness ID: 761

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Free of Pointer not at Start of Buffer zh: -->没有指针不在缓冲区的开始*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序在指向堆上分配的内存资源的指针上调用free（），但指针不在缓冲区的开头。

### Description:

The application calls free() on a pointer to a memory resource that was allocated on the heap, but the pointer is not at the start of the buffer.

### 详细描述:

这可能导致应用程序崩溃，或在某些情况下，修改关键程序变量或执行代码。  
当使用malloc（）系列函数之一显式地在堆上分配内存并且调用free（）时，通常会发生此弱点，但指针算法导致指针位于缓冲区的内部或末尾。

### Extended Description:

This can cause the application to crash, or in some cases, modify critical program variables or execute code.  
This weakness often occurs when the memory is allocated explicitly on the heap with one of the malloc() family functions and free() is called, but pointer arithmetic has caused the pointer to be in the interior or end of the buffer.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

目前，CWE-763是母体，但是可能需要具有非功能特异性的中间母体，类似于CWE-762是CWE-763和CWE-590之间的中间母体的方式。

Weakness ID: 762

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Mismatched Memory Management Routines zh: -->不匹配的内存管理例程*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序尝试将内存资源返回给系统，但它调用的释放函数与最初用于分配该资源的函数不兼容。

### Description:

The application attempts to return a memory resource to the system, but it calls a release function that is not compatible with the function that was originally used to allocate that resource.

### 详细描述:

这种弱点通常可以描述为不匹配的内存管理例程，例如：  
  
  
内存是在堆栈上自动分配的，但它是使用内存管理例程free（）（CWE-590）解除分配的，该程序用于显式分配的堆内存。  
使用一组内存管理函数显式分配内存，并使用不同的集解除分配。例如，可以使用C ++中的malloc（）而不是new运算符分配内存，然后使用delete运算符取消分配。  
  
  
当内存管理功能不匹配时，后果可能与代码执行，内存损坏或程序崩溃一样严重。后果和易用性将根据例程的实现和被管理的对象而有所不同。

### Extended Description:

This weakness can be generally described as mismatching memory management routines, such as:  
  
  
The memory was allocated on the stack (automatically), but it was deallocated using the memory management routine free() (CWE-590), which is intended for explicitly allocated heap memory.  
The memory was allocated explicitly using one set of memory management functions, and deallocated using a different set. For example, memory might be allocated with malloc() in C++ instead of the new operator, and then deallocated with the delete operator.  
  
  
When the memory management functions are mismatched, the consequences may be as severe as code execution, memory corruption, or program crash. Consequences and ease of exploit will vary depending on the implementation of the routines and the object being managed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

任何允许手动管理内存的编程语言都可以实现这种弱点。

Weakness ID: 763

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Release of Invalid Pointer or Reference zh: -->发布无效指针或引用*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序尝试将内存资源返回给系统，但调用错误的释放函数或错误地调用适当的释放函数。

### Description:

The application attempts to return a memory resource to the system, but calls the wrong release function or calls the appropriate release function incorrectly.

### 详细描述:

这种弱点可以采取多种形式，例如：  
  
  
通过一种存储器管理方法明确地或隐含地分配存储器，并使用不同的非兼容功能（CWE-762）解除分配。  
选择的函数调用或内存管理例程是合适的，但它们使用不正确，例如在CWE-761中。

### Extended Description:

This weakness can take several forms, such as:  
  
  
The memory was allocated, explicitly or implicitly, via one memory management method and deallocated using a different, non-compatible function (CWE-762).  
The function calls or memory management routines chosen are appropriate, however they are used incorrectly, such as in CWE-761.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

视图CWE-1000层次结构的这个区域需要额外的工作。可能会在此分支中创建几个条目。目前关注的是内存的free（），但删除和其他相关的发布例程可能需要创建不特定于特定功能的中间条目。此外，其他类型的无效指针的作用，例如过期的指针，即CWE-415 Double Free和释放未初始化的指针，与CWE-457相关。

Weakness ID: 764

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Multiple Locks of a Critical Resource zh: -->关键资源的多重锁定*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件锁定关键资源的次数超过预期，导致系统出现意外状态。

### Description:

The software locks a critical resource more times than intended, leading to an unexpected state in the system.

### 详细描述:

当软件在并发环境中运行并重复锁定关键资源时，后果将根据锁的类型，锁的实现和受保护的资源而有所不同。在某些情况下，例如使用信号量，资源会被合并，额外的锁定调用会减少总可用池的大小，从而可能导致性能下降或拒绝服务。如果这可以由攻击者触发，则它类似于无限制锁（CWE-412）。在二进制锁的上下文中，任何重复的锁定尝试都可能永远不会成功，因为锁已经被保持并且可能无法进行。

### Extended Description:

When software is operating in a concurrent environment and repeatedly locks a critical resource, the consequences will vary based on the type of lock, the lock's implementation, and the resource being protected. In some situations such as with semaphores, the resources are pooled and extra locking calls will reduce the size of the total available pool, possibly leading to degraded performance or a denial of service. If this can be triggered by an attacker, it will be similar to an unrestricted lock (CWE-412). In the context of a binary lock, it is likely that any duplicate locking attempts will never succeed since the lock is already held and progress may not be possible.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

考虑这个弱点的另一种方法是控制流中锁定/解锁次数之间的不平衡。在执行过程中，如果在每个锁定呼叫之后没有在合理的时间内进行后续解锁调用，则如果存在对锁的竞争，则系统性能可能降低或者至少在低于峰值水平的情况下操作。可能需要修改此条目以在将来反映这些概念。

Weakness ID: 765

提交日期 I`m don`t know---> 修改日期 2014-07-30

* **Weakness Name:** *en: --> Multiple Unlocks of a Critical Resource zh: -->多次解锁关键资源*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件解锁关键资源的次数超过预期，导致系统出现意外状态。

### Description:

The software unlocks a critical resource more times than intended, leading to an unexpected state in the system.

### 详细描述:

当软件在并发环境中运行并重复解锁关键资源时，后果将根据锁的类型，锁的实现和受保护的资源而有所不同。在某些情况下，例如信号量，资源被合并，额外的解锁调用将增加可用资源数量的计数，当系统接近容量时可能导致崩溃或不可预测的行为。

### Extended Description:

When software is operating in a concurrent environment and repeatedly unlocks a critical resource, the consequences will vary based on the type of lock, the lock's implementation, and the resource being protected. In some situations such as with semaphores, the resources are pooled and extra calls to unlock will increase the count for the number of available resources, likely resulting in a crash or unpredictable behavior when the system nears capacity.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

考虑这个弱点的另一种方法是控制流中锁定/解锁次数之间的不平衡。在执行过程中，如果在每个锁定呼叫之后没有在合理的时间内进行后续解锁调用，则如果存在对锁的竞争，则系统性能可能降低或者至少在低于峰值水平的情况下操作。可能需要修改此条目以在将来反映这些概念。

Weakness ID: 766

提交日期 2019-01-03---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Critical Data Element Declared Public zh: -->关键数据元素已公开声明*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当预期的安全策略要求其为私有时，软件会声明关键变量，字段或成员是公开的。

### Description:

The software declares a critical variable, field, or member to be public when intended security policy requires it to be private.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 767

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Access to Critical Private Variable via Public Method zh: -->通过公共方法访问关键私有变量*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件定义了一个读取或修改私有变量的公共方法。

### Description:

The software defines a public method that reads or modifies a private variable.

### 详细描述:

如果攻击者修改变量以包含意外值，则可能违反代码其他部分的假设。此外，如果攻击者可以读取私有变量，则可能会泄露敏感信息或使其更容易发起进一步的攻击。

### Extended Description:

If an attacker modifies the variable to contain unexpected values, this could violate assumptions from other parts of the code. Additionally, if an attacker can read the private variable, it may expose sensitive information or make it easier to launch further attacks.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目与公共方法的访问控制密切相关。如果使用适当的访问控制限制公共方法，则私有变量中的信息不会暴露给意外的各方。不正确的访问控制与这个弱点之间可能存在链接或复合关系。

Weakness ID: 768

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incorrect Short Circuit Evaluation zh: -->短路评估不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个带有多个逻辑表达式的条件语句，其中一个非前导表达式可能产生副作用。这可能导致在执行条件之后程序中的意外状态，因为短路逻辑可以防止发生副作用。

### Description:

The software contains a conditional statement with multiple logical expressions in which one of the non-leading expressions may produce side effects. This may lead to an unexpected state in the program after the execution of the conditional, because short-circuiting logic may prevent the side effects from occurring.

### 详细描述:

使用短路评估虽然在C标准中有明确定义，但可能会以引入难以检测的逻辑错误的方式改变控制流，可能在软件执行期间导致错误。如果攻击者可以发现这种不一致，那么获取对系统的任意控制可能是可利用的。  
如果在正常情况下假定“或”语句的第一个条件为真，或者假定“和”语句的第一个条件为假，则任何后续条件都可能包含其自身未检测到的逻辑错误在代码审查或测试期间。  
最后，短路评估的使用可能降低代码的可维护性。

### Extended Description:

Usage of short circuit evaluation, though well-defined in the C standard, may alter control flow in a way that introduces logic errors that are difficult to detect, possibly causing errors later during the software's execution. If an attacker can discover such an inconsistency, it may be exploitable to gain arbitrary control over a system.  
If the first condition of an "or" statement is assumed to be true under normal circumstances, or if the first condition of an "and" statement is assumed to be false, then any subsequent conditional may contain its own logic errors that are not detected during code review or testing.  
Finally, the usage of short circuit evaluation may decrease the maintainability of the code.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 769

提交日期 2017-11-08---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> DEPRECATED: Uncontrolled File Descriptor Consumption zh: -->DEPRECATED：不受控制的文件描述符消耗*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Deprecated*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

此条目已被弃用，因为它与CWE-774重复。所有内容均已转移至CWE-774。

### Description:

This entry has been deprecated because it was a duplicate of CWE-774. All content has been transferred to CWE-774.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 770

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Allocation of Resources Without Limits or Throttling zh: -->无限制或限制的资源分配*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件代表参与者分配可重用资源或资源组，而不会对可分配的资源的大小或数量施加任何限制，这违反了该参与者的预期安全策略。

### Description:

The software allocates a reusable resource or group of resources on behalf of an actor without imposing any restrictions on the size or number of resources that can be allocated, in violation of the intended security policy for that actor.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

“资源枯竭”（CWE-400）目前被视为一个弱点，尽管它更像是一类弱点，都具有相同类型的后果。虽然此条目将CWE-400视为视图1000中的父级，但该关系可能更恰当地描述为链。

Weakness ID: 771

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Missing Reference to Active Allocated Resource zh: -->缺少对活动分配资源的引用*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件未正确维护对已分配资源的引用，这会阻止资源被回收。

### Description:

The software does not properly maintain a reference to a resource that has been allocated, which prevents the resource from being reclaimed.

### 详细描述:

这不一定适用于自动执行垃圾收集的语言或框架，因为删除所有引用可能会充当资源可以回收的信号。

### Extended Description:

This does not necessarily apply in languages or frameworks that automatically perform garbage collection, since the removal of all references may act as a signal that the resource is ready to be reclaimed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 772

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Missing Release of Resource after Effective Lifetime zh: -->有效寿命后缺少资源释放*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件在其有效生命周期结束后，即在不再需要资源之后不释放资源。

### Description:

The software does not release a resource after its effective lifetime has ended, i.e., after the resource is no longer needed.

### 详细描述:

如果资源在使用后未释放，则可能允许攻击者导致拒绝服务。

### Extended Description:

When a resource is not released after use, it can allow attackers to cause a denial of service.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

“资源枯竭”（CWE-400）目前被视为一个弱点，尽管它更像是一类弱点，都具有相同类型的后果。虽然此条目将CWE-400视为视图1000中的父级，但该关系可能更恰当地描述为链。

Weakness ID: 773

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Missing Reference to Active File Descriptor or Handle zh: -->缺少对活动文件描述符或句柄的引用*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件未正确维护对文件描述符或句柄的引用，这会阻止回收文件描述符/句柄。

### Description:

The software does not properly maintain references to a file descriptor or handle, which prevents that file descriptor/handle from being reclaimed.

### 详细描述:

这可能导致软件使用所有可用的文件描述符或句柄，这可能会阻止其他进程执行关键文件处理操作。

### Extended Description:

This can cause the software to consume all available file descriptors or handles, which can prevent other processes from performing critical file processing operations.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 774

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Allocation of File Descriptors or Handles Without Limits or Throttling zh: -->无限制或限制的文件描述符或句柄的分配*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件代表演员分配文件描述符或句柄，而不对可分配的描述符数量施加任何限制，这违反了该演员的预期安全策略。

### Description:

The software allocates file descriptors or handles on behalf of an actor without imposing any restrictions on how many descriptors can be allocated, in violation of the intended security policy for that actor.

### 详细描述:

这可能导致软件使用所有可用的文件描述符或句柄，这可能会阻止其他进程执行关键文件处理操作。

### Extended Description:

This can cause the software to consume all available file descriptors or handles, which can prevent other processes from performing critical file processing operations.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 775

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Missing Release of File Descriptor or Handle after Effective Lifetime zh: -->缺少文件描述符的释放或在有效生命周期后处理*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件在其有效生命周期结束后，即在不再需要文件描述符/句柄之后，不释放文件描述符或句柄。

### Description:

The software does not release a file descriptor or handle after its effective lifetime has ended, i.e., after the file descriptor/handle is no longer needed.

### 详细描述:

如果在使用后未释放文件描述符或句柄（通常通过显式关闭它），攻击者可以通过使用所有可用文件描述符/句柄或以其他方式阻止其他系统进程获取自己的文件描述符/句柄来导致拒绝服务。

### Extended Description:

When a file descriptor or handle is not released after use (typically by explicitly closing it), attackers can cause a denial of service by consuming all available file descriptors/handles, or otherwise preventing other system processes from obtaining their own file descriptors/handles.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 776

提交日期 2013-02-21---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Restriction of Recursive Entity References in DTDs ('XML Entity Expansion') zh: -->DTD中递归实体引用的不正确限制（'XML实体扩展'）*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用XML文档并允许使用文档类型定义（DTD）定义其结构，但它不能正确控制实体的递归定义的数量。

### Description:

The software uses XML documents and allows their structure to be defined with a Document Type Definition (DTD), but it does not properly control the number of recursive definitions of entities.

### 详细描述:

如果DTD包含大量嵌套或递归实体，则在解析时会导致数据爆炸式增长，从而导致拒绝服务。

### Extended Description:

If the DTD contains a large number of nested or recursive entities, this can lead to explosive growth of data when parsed, causing a denial of service.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 777

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Regular Expression without Anchors zh: -->没有锚点的正则表达式*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用正则表达式执行中和，但正则表达式未锚定，可能允许恶意或格式错误的数据泄漏。

### Description:

The software uses a regular expression to perform neutralization, but the regular expression is not anchored and may allow malicious or malformed data to slip through.

### 详细描述:

执行白名单验证等任务时，会检查数据并对其进行修改，以确保数据格式正确并符合安全值列表。如果未锚定正则表达式，则可能在与正则表达式匹配的任何字符串之前或之后包含恶意或格式错误的数据。允许的恶意数据类型取决于应用程序的上下文以及正则表达式中省略的锚点。

### Extended Description:

When performing tasks such as whitelist validation, data is examined and possibly modified to ensure that it is well-formed and adheres to a list of safe values. If the regular expression is not anchored, malicious or malformed data may be included before or after any string matching the regular expression. The type of malicious data that is allowed will depend on the context of the application and which anchors are omitted from the regular expression.

### 问题背景 (Background Detail):

正则表达式通常用于匹配文本模式。在正则表达式中使用锚来指定模式应匹配的位置：在开头，结尾或两者（整个输入）。

### 笔记 (Notes):

没有笔记

Weakness ID: 778

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Insufficient Logging zh: -->记录不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

发生安全关键事件时，软件要么不记录事件，要么在记录事件时忽略有关事件的重要详细信息。

### Description:

When a security-critical event occurs, the software either does not record the event or omits important details about the event when logging it.

### 详细描述:

如果未正确记录安全关键事件（例如登录尝试失败），则可能使恶意行为更难以检测，并可能在攻击成功后阻碍取证分析。

### Extended Description:

When security-critical events are not logged properly, such as a failed login attempt, this can make malicious behavior more difficult to detect and may hinder forensic analysis after an attack succeeds.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 779

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Logging of Excessive Data zh: -->记录过多的数据*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件记录太多信息，使得日志文件难以处理，并可能阻碍恢复工作或攻击后的取证分析。

### Description:

The software logs too much information, making log files hard to process and possibly hindering recovery efforts or forensic analysis after an attack.

### 详细描述:

虽然日志记录是一种很好的做法，并且非常高级别的日志记录适用于调试开发阶段，但生产环境中过多的日志记录可能会妨碍系统管理员检测异常情况的能力。这可以在尝试穿透系统时为攻击者提供掩护，使审计跟踪混乱以进行取证分析，或者使调试生产环境中的问题变得更加困难。

### Extended Description:

While logging is a good practice in general, and very high levels of logging are appropriate for debugging stages of development, too much logging in a production environment might hinder a system administrator's ability to detect anomalous conditions. This can provide cover for an attacker while attempting to penetrate a system, clutter the audit trail for forensic analysis, or make it more difficult to debug problems in a production environment.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 780

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of RSA Algorithm without OAEP zh: -->使用没有OAEP的RSA算法*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用RSA算法，但未包含最佳非对称加密填充（OAEP），这可能会削弱加密。

### Description:

The software uses the RSA algorithm but does not incorporate Optimal Asymmetric Encryption Padding (OAEP), which might weaken the encryption.

### 详细描述:

填充方案通常与加密算法一起使用，以使明文不易预测并使攻击工作复杂化。 OAEP方案通常与RSA一起使用，以消除可预测的通用文本的影响。

### Extended Description:

Padding schemes are often used with cryptographic algorithms to make the plaintext less predictable and complicate attack efforts. The OAEP scheme is often used with RSA to nullify the impact of predictable common text.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目可能有一个与不正确填充相关的新父级，但填充算法中填充的作用可能会有所不同，例如隐藏明文的长度并为密码提供额外的随机位。一般而言，CWE中的加密问题组织不当，需要进一步研究。

Weakness ID: 781

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Address Validation in IOCTL with METHOD\_NEITHER I/O Control Code zh: -->使用METHOD\_NEITHER I / O控制代码在IOCTL中进行不正确的地址验证*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件定义了一个IOCTL，它使用METHOD\_NEITHER进行I / O，但它不验证或错误地验证提供的地址。

### Description:

The software defines an IOCTL that uses METHOD\_NEITHER for I/O, but it does not validate or incorrectly validates the addresses that are provided.

### 详细描述:

当IOCTL使用METHOD\_NEITHER选项进行I / O控制时，IOCTL负责验证已提供给它的地址。如果验证缺失或不正确，攻击者可以提供任意内存地址，从而导致代码执行或拒绝服务。

### Extended Description:

When an IOCTL uses the METHOD\_NEITHER option for I/O control, it is the responsibility of the IOCTL to validate the addresses that have been supplied to it. If validation is missing or incorrect, attackers can supply arbitrary memory addresses, leading to code execution or a denial of service.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

由于IOCTL功能通常执行低级操作并与操作系统密切交互，因此这种弱点可能只出现在用低级语言编写的代码中。

Weakness ID: 782

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Exposed IOCTL with Insufficient Access Control zh: -->暴露的IOCTL访问控制不足*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件实现了一个IOCTL，其功能应该受到限制，但它没有正确实施IOCTL的访问控制。

### Description:

The software implements an IOCTL with functionality that should be restricted, but it does not properly enforce access control for the IOCTL.

### 详细描述:

当IOCTL包含特权功能并且不必要地暴露时，攻击者可以通过调用IOCTL来访问此功能。即使功能是良性的，如果程序员假设IOCTL只能由受信任的进程访问，则可能很少或根本没有验证传入的数据，暴露了如果攻击者无法调用IOCTL则永远无法访问的弱点直。  
IOCTL的实现在操作系统类型和版本之间会有所不同，因此攻击和预防方法可能会有很大差异。

### Extended Description:

When an IOCTL contains privileged functionality and is exposed unnecessarily, attackers may be able to access this functionality by invoking the IOCTL. Even if the functionality is benign, if the programmer has assumed that the IOCTL would only be accessed by a trusted process, there may be little or no validation of the incoming data, exposing weaknesses that would never be reachable if the attacker cannot call the IOCTL directly.  
The implementations of IOCTLs will differ between operating system types and versions, so the methods of attack and prevention may vary widely.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

当程序员假定IOCTL只能由可信方访问时，这可能是许多其他弱点的主要原因。例如，程序或驱动程序可能无法验证Windows环境中的METHOD\_NEITHER IOCTL（CWE-781）中的传入地址，这可能允许缓冲区溢出和类似的攻击发生，即使攻击者永远不应该能够访问IOCTL所有。

Weakness ID: 783

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Operator Precedence Logic Error zh: -->运算符优先逻辑错误*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序使用一个表达式，其中运算符优先级导致使用不正确的逻辑。

### Description:

The program uses an expression in which operator precedence causes incorrect logic to be used.

### 详细描述:

虽然操作符优先级逻辑错误通常只是一个错误，但如果它们用于安全关键代码（例如进行身份验证决策），则会产生严重后果。

### Extended Description:

While often just a bug, operator precedence logic errors can have serious consequences if they are used in security-critical code, such as making an authentication decision.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 784

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Reliance on Cookies without Validation and Integrity Checking in a Security Decision zh: -->依赖于Cookie而无需验证和完整性检查安全决策*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序使用保护机制，该机制依赖于cookie的存在或值，但它不能正确确保cookie对关联用户有效。

### Description:

The application uses a protection mechanism that relies on the existence or values of a cookie, but it does not properly ensure that the cookie is valid for the associated user.

### 详细描述:

攻击者可以在浏览器中轻松修改cookie，也可以在浏览器外部实现客户端代码。攻击者可以通过修改cookie以包含预期值来绕过授权和身份验证等保护机制。

### Extended Description:

Attackers can easily modify cookies, within the browser or by implementing the client-side code outside of the browser. Attackers can bypass protection mechanisms such as authorization and authentication by modifying the cookie to contain an expected value.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可能需要为此条目定义新父级。此条目特定于cookie，反映了2008年和2009年CVE中基于cookie的身份验证报告的大量漏洞。但是，其他类型的输入（如参数或标头）也可用于类似的身份验证或授权。类似的问题（在研究视图下）包括CWE-247和CWE-472。

Weakness ID: 785

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of Path Manipulation Function without Maximum-sized Buffer zh: -->使用没有最大大小缓冲区的路径操作函数*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件调用一个函数来规范化路径或文件名，但它提供的输出缓冲区小于最大可能的大小，例如PATH\_MAX。

### Description:

The software invokes a function for normalizing paths or file names, but it provides an output buffer that is smaller than the maximum possible size, such as PATH\_MAX.

### 详细描述:

将不适当大小的输出缓冲区传递给路径操作函数可能会导致缓冲区溢出。这些函数包括realpath（），readlink（），PathAppend（）等。

### Extended Description:

Passing an inadequately-sized output buffer to a path manipulation function can result in a buffer overflow. Such functions include realpath(), readlink(), PathAppend(), and others.

### 问题背景 (Background Detail):

Windows提供了大量实用程序函数来处理包含文件名的缓冲区。在大多数情况下，结果将在作为输入传入的缓冲区中返回。 （通常会对文件名进行修改。）大多数函数要求缓冲区的长度至少为MAX\_PATH字节，但是您应该单独检查每个函数的文档。如果缓冲区不足以存储操作结果，则可能发生缓冲区溢出。

### 笔记 (Notes):

这个条目的大部分原本是CWE-249的一部分，由于几个原因而被弃用。

Weakness ID: 786

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Access of Memory Location Before Start of Buffer zh: -->缓冲区启动前访问内存位置*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件使用索引或指针读取或写入缓冲区，索引或指针在缓冲区开始之前引用存储器位置。

### Description:

The software reads or writes to a buffer using an index or pointer that references a memory location prior to the beginning of the buffer.

### 详细描述:

这通常发生在指针或其索引递减到缓冲区之前的位置，指针算术导致有效内存位置开始之前的位置或使用负索引时。

### Extended Description:

This typically occurs when a pointer or its index is decremented to a position before the buffer, when pointer arithmetic results in a position before the beginning of the valid memory location, or when a negative index is used.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 787

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Out-of-bounds Write zh: -->越界写作*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件将数据写入预期缓冲区的结束或开始之前。

### Description:

The software writes data past the end, or before the beginning, of the intended buffer.

### 详细描述:

通常，这可能导致数据损坏，崩溃或代码执行。软件可以修改索引或执行引用超出缓冲区边界的存储器位置的指针算术。随后的写操作会产生未定义或意外的结果。

### Extended Description:

Typically, this can result in corruption of data, a crash, or code execution. The software may modify an index or perform pointer arithmetic that references a memory location that is outside of the boundaries of the buffer. A subsequent write operation then produces undefined or unexpected results.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 788

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Access of Memory Location After End of Buffer zh: -->缓冲区结束后访问内存位置*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件使用索引或指针读取或写入缓冲区，该索引或指针在缓冲区结束后引用内存位置。

### Description:

The software reads or writes to a buffer using an index or pointer that references a memory location after the end of the buffer.

### 详细描述:

这通常发生在指针或其索引递减到缓冲区之前的位置时;当指针算术结果在缓冲区之前的位置时;或者当使用负索引时，它会在缓冲区之前生成一个位置。

### Extended Description:

This typically occurs when a pointer or its index is decremented to a position before the buffer; when pointer arithmetic results in a position before the buffer; or when a negative index is used, which generates a position before the buffer.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 789

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Uncontrolled Memory Allocation zh: -->不受控制的内存分配*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品根据不受信任的大小值分配内存，但它不验证或错误地验证大小，允许分配任意数量的内存。

### Description:

The product allocates memory based on an untrusted size value, but it does not validate or incorrectly validates the size, allowing arbitrary amounts of memory to be allocated.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这个弱点可能与整数溢出密切相关（CWE-190）。整数溢出攻击将集中于提供一个极大的数字，触发溢出，导致分配的内存少于预期。通过提供不会触发整数溢出的大值，攻击者仍可能导致分配过多的内存。

Weakness ID: 790

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Filtering of Special Elements zh: -->特殊元素的过滤不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件从上游组件接收数据，但在将特殊元素发送到下游组件之前不会过滤或错误过滤特殊元素。

### Description:

The software receives data from an upstream component, but does not filter or incorrectly filters special elements before sending it to a downstream component.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 791

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incomplete Filtering of Special Elements zh: -->特殊元素的不完全过滤*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件从上游组件接收数据，但在将特殊元素发送到下游组件之前不会完全过滤特殊元素。

### Description:

The software receives data from an upstream component, but does not completely filter special elements before sending it to a downstream component.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 792

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incomplete Filtering of One or More Instances of Special Elements zh: -->一个或多个特殊元素实例的不完全过滤*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件从上游组件接收数据，但在将特定元素的一个或多个实例发送到下游组件之前不会完全过滤它们。

### Description:

The software receives data from an upstream component, but does not completely filter one or more instances of special elements before sending it to a downstream component.

### 详细描述:

这种性质的不完全过滤包括：  
  
  
仅当存在更多时，才过滤特殊元素的单个实例，或者  
不过滤所有实例或存在多个特殊元素的所有元素。

### Extended Description:

Incomplete filtering of this nature involves either:  
  
  
only filtering a single instance of a special element when more exist, or  
not filtering all instances or all elements where multiple special elements exist.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 793

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Only Filtering One Instance of a Special Element zh: -->仅过滤特殊元素的一个实例*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件从上游组件接收数据，但仅在将特定元素的单个实例发送到下游组件之前对其进行过滤。

### Description:

The software receives data from an upstream component, but only filters a single instance of a special element before sending it to a downstream component.

### 详细描述:

这种性质的不完全过滤可能是位置相关的，因为仅过滤了第一个或最后一个元素。

### Extended Description:

Incomplete filtering of this nature may be location-dependent, as in only the first or last element is filtered.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 794

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incomplete Filtering of Multiple Instances of Special Elements zh: -->多特殊元素实例的不完全过滤*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件从上游组件接收数据，但在将特定元素发送到下游组件之前不会过滤所有特定元素的实例。

### Description:

The software receives data from an upstream component, but does not filter all instances of a special element before sending it to a downstream component.

### 详细描述:

这种性质的不完全过滤可能适用于：  
  
  
顺序元素（彼此相邻的特殊元素）或  
非顺序元素（在不同位置多次出现的特殊元素）。

### Extended Description:

Incomplete filtering of this nature may be applied to:  
  
  
sequential elements (special elements that appear next to each other) or  
non-sequential elements (special elements that appear multiple times in different locations).

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 795

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Only Filtering Special Elements at a Specified Location zh: -->仅在指定位置过滤特殊元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件从上游组件接收数据，但仅考虑指定位置的特殊元素，从而丢失在将其发送到下游组件之前可能存在的剩余特殊元素。

### Description:

The software receives data from an upstream component, but only accounts for special elements at a specified location, thereby missing remaining special elements that may exist before sending it to a downstream component.

### 详细描述:

过滤器可能仅在特殊元素出现时考虑它们的实例：  
  
  
相对于标记（例如“在字符串的开头/结尾;第二个参数”），或  
在绝对位置（例如“字节数10”）。  
  
  
这可能会在数据中留下与过滤器位置不匹配的特殊元素，但仍可能存在危险。

### Extended Description:

A filter might only account for instances of special elements when they occur:  
  
  
relative to a marker (e.g. "at the beginning/end of string; the second argument"), or  
at an absolute position (e.g. "byte number 10").  
  
  
This may leave special elements in the data that did not match the filter position, but still may be dangerous.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 796

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Only Filtering Special Elements Relative to a Marker zh: -->仅过滤相对于标记的特殊元素*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件从上游组件接收数据，但仅考虑相对于标记定位的特殊元素（例如“在字符串的开头/结尾;第二个参数”），从而丢失在将其发送到之前可能存在的剩余特殊元素下游组件。

### Description:

The software receives data from an upstream component, but only accounts for special elements positioned relative to a marker (e.g. "at the beginning/end of a string; the second argument"), thereby missing remaining special elements that may exist before sending it to a downstream component.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 797

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Only Filtering Special Elements at an Absolute Position zh: -->仅在绝对位置过滤特殊元素*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件从上游组件接收数据，但仅考虑绝对位置处的特殊元素（例如“字节数10”），从而丢失在将其发送到下游组件之前可能存在的剩余特殊元素。

### Description:

The software receives data from an upstream component, but only accounts for special elements at an absolute position (e.g. "byte number 10"), thereby missing remaining special elements that may exist before sending it to a downstream component.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 798

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Hard-coded Credentials zh: -->使用硬编码凭证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含硬编码凭证，例如密码或加密密钥，用于其自身的入站身份验证，与外部组件的出站通信或内部数据的加密。

### Description:

The software contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.

### 详细描述:

硬编码凭证通常会创建一个重要漏洞，允许攻击者绕过软件管理员配置的身份验证。系统管理员可能难以检测到此漏洞。即使被检测到，也很难修复，因此管理员可能被迫完全禁用该产品。主要有两种变化：  
  
入站：该软件包含一种身份验证机制，可根据硬编码凭据集检查输入凭据。  
出站：软件连接到另一个系统或组件，它包含用于连接到该组件的硬编码凭据。  
  
在入站变体中，将创建默认管理帐户，并将简单密码硬编码到产品中并与该帐户关联。此硬编码密码对于产品的每次安装都是相同的，并且系统管理员通常无法在不手动修改程序或修补软件的情况下更改或禁用密码。如果密码被发现或发布（在Internet上很常见），那么任何知道此密码的人都可以访问该产品。最后，由于软件的所有安装都具有相同的密码，即使在不同的组织中，也可以实现诸如蠕虫之类的大规模攻击。  
Outbound变体适用于使用后端服务进行身份验证的前端系统。后端服务可能需要一个可以轻松发现的固定密码。程序员可以简单地将这些后端凭证硬编码到前端软件中。该程序的任何用户都可以提取密码。具有硬编码密码的客户端系统构成了更大的威胁，因为从二进制文件中提取密码通常非常简单。

### Extended Description:

Hard-coded credentials typically create a significant hole that allows an attacker to bypass the authentication that has been configured by the software administrator. This hole might be difficult for the system administrator to detect. Even if detected, it can be difficult to fix, so the administrator may be forced into disabling the product entirely. There are two main variations:  
  
Inbound: the software contains an authentication mechanism that checks the input credentials against a hard-coded set of credentials.  
Outbound: the software connects to another system or component, and it contains hard-coded credentials for connecting to that component.  
  
In the Inbound variant, a default administration account is created, and a simple password is hard-coded into the product and associated with that account. This hard-coded password is the same for each installation of the product, and it usually cannot be changed or disabled by system administrators without manually modifying the program, or otherwise patching the software. If the password is ever discovered or published (a common occurrence on the Internet), then anybody with knowledge of this password can access the product. Finally, since all installations of the software will have the same password, even across different organizations, this enables massive attacks such as worms to take place.  
The Outbound variant applies to front-end systems that authenticate with a back-end service. The back-end service may require a fixed password which can be easily discovered. The programmer may simply hard-code those back-end credentials into the front-end software. Any user of that program may be able to extract the password. Client-side systems with hard-coded passwords pose even more of a threat, since the extraction of a password from a binary is usually very simple.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 799

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Control of Interaction Frequency zh: -->交互频率控制不当*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件没有正确限制它与演员的交互次数或频率，例如传入请求的数量。

### Description:

The software does not properly limit the number or frequency of interactions that it has with an actor, such as the number of incoming requests.

### 详细描述:

这可以允许演员比预期更频繁地执行动作。演员可以是人或自动过程，例如病毒或机器人。这可能会导致拒绝服务，破坏程序逻辑（例如限制人们进行单一投票）或其他后果。例如，身份验证例程可能不会限制攻击者猜测密码的次数。或者，一个网站可能会进行民意调查，但只希望人们每天最多投票一次。

### Extended Description:

This can allow the actor to perform actions more frequently than expected. The actor could be a human or an automated process such as a virus or bot. This could be used to cause a denial of service, compromise program logic (such as limiting humans to a single vote), or other consequences. For example, an authentication routine might not limit the number of times an attacker can guess a password. Or, a web site might conduct a poll but only expect humans to vote a maximum of once a day.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 804

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Guessable CAPTCHA zh: -->猜猜CAPTCHA*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用CAPTCHA挑战，但挑战可以被非人类演员猜测或自动识别。

### Description:

The software uses a CAPTCHA challenge, but the challenge can be guessed or automatically recognized by a non-human actor.

### 详细描述:

自动攻击者可以绕过对CAPTCHA挑战的预期保护，并以比人为可能更高的频率执行操作，例如发起垃圾邮件攻击。  
可猜测的CAPTCHA可能有几种不同的原因：  
  
  
未经模糊处理的源图像没有足够失真的音频或视频图像。  
生成的问题是可以自动识别的格式，例如数学问题。  
可能答案数量有限的问题，例如出生年份或喜爱的运动队。  
可以使用数据库访问答案的一般知识或琐事问题，例如国家首都或受欢迎的演员。  
与CAPTCHA相关联的其他数据可以提供关于其内容的提示，例如其文件名包含在CAPTCHA中使用的单词的图像。

### Extended Description:

An automated attacker could bypass the intended protection of the CAPTCHA challenge and perform actions at a higher frequency than humanly possible, such as launching spam attacks.  
There can be several different causes of a guessable CAPTCHA:  
  
  
An audio or visual image that does not have sufficient distortion from the unobfuscated source image.  
A question is generated that with a format that can be automatically recognized, such as a math question.  
A question for which the number of possible answers is limited, such as birth years or favorite sports teams.  
A general-knowledge or trivia question for which the answer can be accessed using a data base, such as country capitals or popular actors.  
Other data associated with the CAPTCHA may provide hints about its contents, such as an image whose filename contains the word that is used in the CAPTCHA.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 805

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Buffer Access with Incorrect Length Value zh: -->缓冲区访问长度值不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用顺序操作来读取或写入缓冲区，但它使用不正确的长度值，使其访问超出缓冲区范围的内存。

### Description:

The software uses a sequential operation to read or write a buffer, but it uses an incorrect length value that causes it to access memory that is outside of the bounds of the buffer.

### 详细描述:

当长度值超过目标大小时，可能会发生缓冲区溢出。

### Extended Description:

When the length value exceeds the size of the destination, a buffer overflow could occur.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 806

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Buffer Access Using Size of Source Buffer zh: -->使用源缓冲区大小的缓冲区访问*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在读取或写入目标缓冲区时，软件使用源缓冲区的大小，这可能导致它访问超出缓冲区范围的内存。

### Description:

The software uses the size of a source buffer when reading from or writing to a destination buffer, which may cause it to access memory that is outside of the bounds of the buffer.

### 详细描述:

当目标的大小小于源的大小时，可能会发生缓冲区溢出。

### Extended Description:

When the size of the destination is smaller than the size of the source, a buffer overflow could occur.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 807

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Reliance on Untrusted Inputs in a Security Decision zh: -->依赖于安全决策中的不受信任的输入*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序使用依赖于输入的存在或值的保护机制，但输入可以由不可信的actor以绕过保护机制的方式修改。

### Description:

The application uses a protection mechanism that relies on the existence or values of an input, but the input can be modified by an untrusted actor in a way that bypasses the protection mechanism.

### 详细描述:

开发人员可能会认为不能修改cookie，环境变量和隐藏表单字段等输入。但是，攻击者可以使用自定义客户端或其他攻击来更改这些输入。可能无法检测到此更改。当根据这些输入的值进行身份验证和授权等安全决策时，攻击者可以绕过软件的安全性。  
如果没有足够的加密，完整性检查或其他机制，则无法信任源自局外人的任何输入。

### Extended Description:

Developers may assume that inputs such as cookies, environment variables, and hidden form fields cannot be modified. However, an attacker could change these inputs using customized clients or other attacks. This change might not be detected. When security decisions such as authentication and authorization are made based on the values of these inputs, attackers can bypass the security of the software.  
Without sufficient encryption, integrity checking, or other mechanism, any input that originates from an outsider cannot be trusted.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 820

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Missing Synchronization zh: -->缺少同步*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件以并发方式使用共享资源，但不尝试同步对资源的访问。

### Description:

The software utilizes a shared resource in a concurrent manner but does not attempt to synchronize access to the resource.

### 详细描述:

如果未同步对共享资源的访问，则资源可能不处于软件期望的状态。这可能会导致意外或不安全的行为，尤其是在攻击者可以影响共享资源的情况下。

### Extended Description:

If access to a shared resource is not synchronized, then the resource may not be in a state that is expected by the software. This might lead to unexpected or insecure behaviors, especially if an attacker can influence the shared resource.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 821

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incorrect Synchronization zh: -->同步不正确*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件以并发方式使用共享资源，但它无法正确同步对资源的访问。

### Description:

The software utilizes a shared resource in a concurrent manner but it does not correctly synchronize access to the resource.

### 详细描述:

如果未正确同步对共享资源的访问，则资源可能不处于软件期望的状态。这可能会导致意外或不安全的行为，尤其是在攻击者可以影响共享资源的情况下。

### Extended Description:

If access to a shared resource is not correctly synchronized, then the resource may not be in a state that is expected by the software. This might lead to unexpected or insecure behaviors, especially if an attacker can influence the shared resource.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 822

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Untrusted Pointer Dereference zh: -->不受信任的指针解除引用*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

程序从不受信任的源获取值，将此值转换为指针，并取消引用结果指针。

### Description:

The program obtains a value from an untrusted source, converts this value to a pointer, and dereferences the resulting pointer.

### 详细描述:

攻击者可以为程序不期望的内存位置提供指针。如果指针被解除引用以进行写操作，则攻击可能允许修改关键程序状态变量，导致崩溃或执行代码。如果取消引用操作用于读取，则攻击可能允许读取敏感数据，导致崩溃或将程序变量设置为意外值（因为该值将从意外的内存位置读取）。  
这种弱点有几种变体，包括但不一定限于：  
  
  
不受信任的值直接作为函数调用调用。  
在操作系统内核或驱动程序中，“userland”和特权内存空间之间存在边界，不受信任的指针可能通过API或系统调用进入（有关此类示例，请参阅CWE-781）。  
当不必接受输入时，无意中接受来自不受控制的控制范围的值。当代码最初开发为由非网络环境中的单个用户运行时，可能会发生这种情况，然后将代码移植到网络环境或以其他方式暴露给网络环境。

### Extended Description:

An attacker can supply a pointer for memory locations that the program is not expecting. If the pointer is dereferenced for a write operation, the attack might allow modification of critical program state variables, cause a crash, or execute code. If the dereferencing operation is for a read, then the attack might allow reading of sensitive data, cause a crash, or set a program variable to an unexpected value (since the value will be read from an unexpected memory location).  
There are several variants of this weakness, including but not necessarily limited to:  
  
  
The untrusted value is directly invoked as a function call.  
In OS kernels or drivers where there is a boundary between "userland" and privileged memory spaces, an untrusted pointer might enter through an API or system call (see CWE-781 for one such example).  
Inadvertently accepting the value from an untrusted control sphere when it did not have to be accepted as input at all. This might occur when the code was originally developed to be run by a single user in a non-networked environment, and the code is then ported to or otherwise exposed to a networked environment.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

不正确的指针解引用与缓冲操作相关的其他弱点之间存在密切关系。关于这些关系可能没有足够的社区协议。需要进一步研究以确定这些关系何时是链，复合材料，透视/分层或其他类型的关系。截至2010年9月，大多数关系被捕获为链条。

Weakness ID: 823

提交日期 I`m don`t know---> 修改日期 2012-05-11

* **Weakness Name:** *en: --> Use of Out-of-range Pointer Offset zh: -->使用超出范围的指针偏移*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

程序对有效指针执行指针运算，但它使用的偏移量可以指向结果指针的有效存储器位置的预期范围之外。

### Description:

The program performs pointer arithmetic on a valid pointer, but it uses an offset that can point outside of the intended range of valid memory locations for the resulting pointer.

### 详细描述:

虽然指针可以包含对任意存储器位置的引用，但是程序通常仅打算使用指针来访问存储器的有限部分，例如用于访问单个阵列的连续存储器。  
程序可以使用偏移量来访问存储在结构化数据中的字段或子元素。如果偏移来自不受信任的来源，是计算错误的结果，或者由于另一个错误而发生，则偏移可能超出范围。  
如果攻击者可以控制或影响偏移量，使其指向结构的预期边界之外，则攻击者可能能够读取或写入程序中其他地方使用的内存位置。因此，攻击可能会改变通过程序变量访问的软件状态，导致崩溃或不稳定的行为，并可能导致代码执行。

### Extended Description:

While a pointer can contain a reference to any arbitrary memory location, a program typically only intends to use the pointer to access limited portions of memory, such as contiguous memory used to access an individual array.  
Programs may use offsets in order to access fields or sub-elements stored within structured data. The offset might be out-of-range if it comes from an untrusted source, is the result of an incorrect calculation, or occurs because of another error.  
If an attacker can control or influence the offset so that it points outside of the intended boundaries of the structure, then the attacker may be able to read or write to memory locations that are used elsewhere in the program. As a result, the attack might change the state of the software as accessed through program variables, cause a crash or instable behavior, and possibly lead to code execution.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

不正确的指针解引用与缓冲操作相关的其他弱点之间存在密切关系。关于这些关系可能没有足够的社区协议。需要进一步研究以确定这些关系何时是链，复合材料，透视/分层或其他类型的关系。截至2010年9月，大多数关系被捕获为链条。

Weakness ID: 824

提交日期 I`m don`t know---> 修改日期 2015-12-07

* **Weakness Name:** *en: --> Access of Uninitialized Pointer zh: -->访问未初始化的指针*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

程序访问或使用尚未初始化的指针。

### Description:

The program accesses or uses a pointer that has not been initialized.

### 详细描述:

如果指针包含未初始化的值，则该值可能不指向有效的内存位置。这可能导致程序读取或写入意外的内存位置，从而导致拒绝服务。如果未初始化的指针用作函数调用，则可以调用任意函数。如果攻击者可以影响指针中包含的未初始化内存部分，则可以利用此弱点执行代码或执行其他攻击。  
根据内存布局，相关的内存管理行为和程序操作，攻击者可能能够影响未初始化指针的内容，从而获得对要访问的内存位置的更精细控制。

### Extended Description:

If the pointer contains an uninitialized value, then the value might not point to a valid memory location. This could cause the program to read from or write to unexpected memory locations, leading to a denial of service. If the uninitialized pointer is used as a function call, then arbitrary functions could be invoked. If an attacker can influence the portion of uninitialized memory that is contained in the pointer, this weakness could be leveraged to execute code or perform other attacks.  
Depending on memory layout, associated memory management behaviors, and program operation, the attacker might be able to influence the contents of the uninitialized pointer, thus gaining more fine-grained control of the memory location to be accessed.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

不正确的指针解引用与缓冲操作相关的其他弱点之间存在密切关系。关于这些关系可能没有足够的社区协议。需要进一步研究以确定这些关系何时是链，复合材料，透视/分层或其他类型的关系。截至2010年9月，大多数关系被捕获为链条。

Weakness ID: 825

提交日期 I`m don`t know---> 修改日期 2013-02-21

* **Weakness Name:** *en: --> Expired Pointer Dereference zh: -->已过期指针解除引用*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序取消引用一个指针，该指针包含以前有效但不再有效的内存位置。

### Description:

The program dereferences a pointer that contains a location for memory that was previously valid, but is no longer valid.

### 详细描述:

当程序释放内存但它保持指向该内存的指针时，可能会在以后重新分配内存。如果访问原始指针以读取或写入数据，则可能导致程序读取或修改由不同功能或进程使用的数据。根据新分配的内存的使用方式，这可能会导致拒绝服务，信息泄露或代码执行。

### Extended Description:

When a program releases memory, but it maintains a pointer to that memory, then the memory might be re-allocated at a later time. If the original pointer is accessed to read or write data, then this could cause the program to read or modify data that is in use by a different function or process. Depending on how the newly-allocated memory is used, this could lead to a denial of service, information exposure, or code execution.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

不正确的指针解引用与缓冲操作相关的其他弱点之间存在密切关系。关于这些关系可能没有足够的社区协议。需要进一步研究以确定这些关系何时是链，复合材料，透视/分层或其他类型的关系。截至2010年9月，大多数关系被捕获为链条。

Weakness ID: 826

提交日期 I`m don`t know---> 修改日期 2011-06-01

* **Weakness Name:** *en: --> Premature Release of Resource During Expected Lifetime zh: -->在预期的生命周期内过早释放资源*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序释放一个仍然打算由程序本身或另一个actor使用的资源。

### Description:

The program releases a resource that is still intended to be used by the program itself or another actor.

### 详细描述:

这个弱点集中在程序不应该释放资源但是无论如何都要执行释放的错误。这与程序在适当的时间释放资源的弱点不同，但它维护对资源的引用，稍后它将访问该资源。对于这个弱点，资源在后续访问时仍应有效。  
当程序释放仍在使用的资源时，可能仍将对此资源进行操作，这可能在此期间被重新利用，导致类似于CWE-825的问题。后果可能包括拒绝服务，信息泄露或代码执行。

### Extended Description:

This weakness focuses on errors in which the program should not release a resource, but performs the release anyway. This is different than a weakness in which the program releases a resource at the appropriate time, but it maintains a reference to the resource, which it later accesses. For this weaknesses, the resource should still be valid upon the subsequent access.  
When a program releases a resource that is still being used, it is possible that operations will still be taken on this resource, which may have been repurposed in the meantime, leading to issues similar to CWE-825. Consequences may include denial of service, information exposure, or code execution.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

截至2010年9月，研究不足和报告不足。尽管重点主要放在内存分配和解除分配上，但在高可见性软件中已经报告了这一弱点。很少有这种弱点的例子与内存管理没有直接关系，尽管这些弱点很可能发生在其他类型资源的真实软件中。

Weakness ID: 827

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Control of Document Type Definition zh: -->文件类型定义控制不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不会将对文档类型定义（DTD）的引用限制为预期的控制范围。这可能允许攻击者引用任意DTD，可能导致软件暴露文件，消耗过多的系统资源，或代表攻击者执行任意http请求。

### Description:

The software does not restrict a reference to a Document Type Definition (DTD) to the intended control sphere. This might allow attackers to reference arbitrary DTDs, possibly causing the software to expose files, consume excessive system resources, or execute arbitrary http requests on behalf of the attacker.

### 详细描述:

在处理DTD时，他们可能会尝试在执行解析的计算机上读取或包含文件。如果攻击者能够控制DTD，则攻击者可能能够指定敏感资源或请求或提供恶意内容。  
例如，SOAP规范禁止SOAP消息包含DTD。

### Extended Description:

As DTDs are processed, they might try to read or include files on the machine performing the parsing. If an attacker is able to control the DTD, then the attacker might be able to specify sensitive resources or requests or provide malicious content.  
For example, the SOAP specification prohibits SOAP messages from containing DTDs.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 828

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Signal Handler with Functionality that is not Asynchronous-Safe zh: -->具有非异步安全功能的信号处理程序*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件定义了一个信号处理程序，它包含非异步安全的代码序列，即功能不可重入，或者可以被中断。

### Description:

The software defines a signal handler that contains code sequences that are not asynchronous-safe, i.e., the functionality is not reentrant, or it can be interrupted.

### 详细描述:

这可能导致意外的系统状态，具有各种潜在后果，具体取决于上下文，包括拒绝服务和代码执行。  
信号处理器通常用于中断程序的正常功能，或甚至其他信号，以便通知事件的过程。当信号处理程序使用全局或静态变量，或调用最终依赖于此类状态或其关联元数据的函数时，它可能会破坏正常功能正在使用的系统状态。这可能使程序受到竞争条件或允许攻击者导致程序状态被破坏的其他弱点。虽然拒绝服务通常是后果，但在某些情况下，可以利用这种弱点来执行代码。  
有几种不同的方案可以引入此问题：  
  
  
从处理程序中调用非重入函数。一个例子是malloc（），它在管理内存时修改内部全局变量。很少有函数实际上是可重入的。  
代码序列（不一定是函数调用）包含全局变量或相关元数据或结构的非原子使用，可由程序的其他功能（包括其他信号处理程序）访问。通常，注册相同的功能来处理多个信号。  
信号处理函数最多只能运行一次，但可以多次调用。这可能通过重复传递相同的信号，或通过传递具有相同处理函数的不同信号（CWE-831）来实现。  
  
  
请注意，在某些环境或上下文中，信号处理程序可能会自行中断。  
如果信号处理程序和软件的正常行为都必须对同一组状态变量进行操作，并且在正常执行对这些变量的修改过程中接收到信号，则变量可能处于不正确或损坏状态在信号处理程序执行期间，返回时可能仍然不正确或损坏。

### Extended Description:

This can lead to an unexpected system state with a variety of potential consequences depending on context, including denial of service and code execution.  
Signal handlers are typically intended to interrupt normal functionality of a program, or even other signals, in order to notify the process of an event. When a signal handler uses global or static variables, or invokes functions that ultimately depend on such state or its associated metadata, then it could corrupt system state that is being used by normal functionality. This could subject the program to race conditions or other weaknesses that allow an attacker to cause the program state to be corrupted. While denial of service is frequently the consequence, in some cases this weakness could be leveraged for code execution.  
There are several different scenarios that introduce this issue:  
  
  
Invocation of non-reentrant functions from within the handler. One example is malloc(), which modifies internal global variables as it manages memory. Very few functions are actually reentrant.  
Code sequences (not necessarily function calls) contain non-atomic use of global variables, or associated metadata or structures, that can be accessed by other functionality of the program, including other signal handlers. Frequently, the same function is registered to handle multiple signals.  
The signal handler function is intended to run at most one time, but instead it can be invoked multiple times. This could happen by repeated delivery of the same signal, or by delivery of different signals that have the same handler function (CWE-831).  
  
  
Note that in some environments or contexts, it might be possible for the signal handler to be interrupted itself.  
If both a signal handler and the normal behavior of the software have to operate on the same set of state variables, and a signal is received in the middle of the normal execution's modifications of those variables, the variables may be in an incorrect or corrupt state during signal handler execution, and possibly still incorrect or corrupt upon return.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 829

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Inclusion of Functionality from Untrusted Control Sphere zh: -->从不受信任的控制领域中包含功能*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件从预期控制范围之外的源导入，要求或包含可执行功能（例如库）。

### Description:

The software imports, requires, or includes executable functionality (such as a library) from a source that is outside of the intended control sphere.

### 详细描述:

当包括第三方功能（例如Web小部件，库或其他功能源）时，软件必须有效地信任该功能。如果没有足够的保护机制，该功能本质上可能是恶意的（无论是来自不受信任的来源，被欺骗，还是来自受信任来源的传输中的修改）。该功能还可能包含其自身的弱点，或授予对应保持对基本系统私有的其他功能和状态信息的访问权限，例如系统状态信息，敏感应用程序数据或Web应用程序的DOM。  
这可能会导致许多不同的后果，具体取决于所包含的功能，但一些示例包括注入恶意软件，通过向不受信任的功能授予过多权限或权限的信息暴露，基于DOM的XSS漏洞，窃取用户的cookie或打开重定向到恶意软件（ CWE-601）。

### Extended Description:

When including third-party functionality, such as a web widget, library, or other source of functionality, the software must effectively trust that functionality. Without sufficient protection mechanisms, the functionality could be malicious in nature (either by coming from an untrusted source, being spoofed, or being modified in transit from a trusted source). The functionality might also contain its own weaknesses, or grant access to additional functionality and state information that should be kept private to the base system, such as system state information, sensitive application data, or the DOM of a web application.  
This might lead to many different consequences depending on the included functionality, but some examples include injection of malware, information exposure by granting excessive privileges or permissions to the untrusted functionality, DOM-based XSS vulnerabilities, stealing user's cookies, or open redirect to malware (CWE-601).

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 830

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Inclusion of Web Functionality from an Untrusted Source zh: -->从不受信任的来源中包含Web功能*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包括来自另一个域的Web功能（例如Web小部件），这使得它在软件的域内操作，可能授予对不可信源的软件的完全访问和控制。

### Description:

The software includes web functionality (such as a web widget) from another domain, which causes it to operate within the domain of the software, potentially granting total access and control of the software to the untrusted source.

### 详细描述:

在基于Web的环境中包含第三方功能是有风险的，尤其是在功能源不受信任的情况下。  
即使第三方是受信任的来源，如果受信任的来源受到损害，或者如果代码在从第三方传输到软件时被修改，则软件仍可能受到攻击和恶意行为。  
这种弱点在Web上的“mashup”开发中很常见，其中可能包括来自其他域的源功能。例如，可以使用“<SCRIPT SRC =”http://other.domain.here“>”标记插入基于Javascript的Web小部件，这会导致代码在软件的域中运行，而不是远程站点从中加载小部件。因此，包含的代码可以访问本地DOM，包括开发人员可能不希望远程站点能够访问的cookie和其他数据。  
这种依赖性可能是期望的，甚至是需要的，但有时程序员并不知道存在依赖性。

### Extended Description:

Including third party functionality in a web-based environment is risky, especially if the source of the functionality is untrusted.  
Even if the third party is a trusted source, the software may still be exposed to attacks and malicious behavior if that trusted source is compromised, or if the code is modified in transmission from the third party to the software.  
This weakness is common in "mashup" development on the web, which may include source functionality from other domains. For example, Javascript-based web widgets may be inserted by using '<SCRIPT SRC="http://other.domain.here">' tags, which causes the code to run in the domain of the software, not the remote site from which the widget was loaded. As a result, the included code has access to the local DOM, including cookies and other data that the developer might not want the remote site to be able to access.  
Such dependencies may be desirable, or even required, but sometimes programmers are not aware that a dependency exists.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 831

提交日期 I`m don`t know---> 修改日期 2014-06-23

* **Weakness Name:** *en: --> Signal Handler Function Associated with Multiple Signals zh: -->与多个信号相关的信号处理器功能*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件定义了一个函数，用作多个信号的处理程序。

### Description:

The software defines a function that is used as a handler for more than one signal.

### 详细描述:

虽然有时是有意和安全的，但是当使用相同的函数来处理多个信号时，如果函数使用其本地声明之外的任何状态（例如全局变量或非重入函数）或者具有任何副作用，则可能发生竞争条件。  
攻击者可以发送一个调用处理函数的信号;在许多操作系统中，这通常会阻止相同的信号再次调用处理程序，至少在处理程序函数完成执行之前。但是，攻击者可以发送与同一处理函数关联的不同信号。这可能会在原始处理函数仍在执行时中断它。如果存在共享状态，则状态可能已损坏。这可能会导致各种潜在后果，具体取决于上下文，包括拒绝服务和代码执行。  
当信号处理程序仅被设计为执行一次（如果有的话）时，会出现另一种很少探索的可能性。通过发送多个信号，攻击者可以多次调用该函数。这可能会产生额外的，无意的副作用。甚至可能不需要竞争条件;攻击者可以发送一个信号，等待它被处理，然后发送另一个信号。

### Extended Description:

While sometimes intentional and safe, when the same function is used to handle multiple signals, a race condition could occur if the function uses any state outside of its local declaration, such as global variables or non-reentrant functions, or has any side effects.  
An attacker could send one signal that invokes the handler function; in many OSes, this will typically prevent the same signal from invoking the handler again, at least until the handler function has completed execution. However, the attacker could then send a different signal that is associated with the same handler function. This could interrupt the original handler function while it is still executing. If there is shared state, then the state could be corrupted. This can lead to a variety of potential consequences depending on context, including denial of service and code execution.  
Another rarely-explored possibility arises when the signal handler is only designed to be executed once (if at all). By sending multiple signals, an attacker could invoke the function more than once. This may generate extra, unintended side effects. A race condition might not even be necessary; the attacker could send one signal, wait until it is handled, then send the other signal.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 832

提交日期 I`m don`t know---> 修改日期 2011-06-01

* **Weakness Name:** *en: --> Unlock of a Resource that is not Locked zh: -->解锁未锁定的资源*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件尝试解锁未锁定的资源。

### Description:

The software attempts to unlock a resource that is not locked.

### 详细描述:

根据锁定功能，解锁非锁定资源可能会导致内存损坏或对资源（或其关联的元数据用于跟踪锁定）的其他修改。

### Extended Description:

Depending on the locking functionality, an unlock of a non-locked resource might cause memory corruption or other modification to the resource (or its associated metadata that is used for tracking locks).

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 833

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Deadlock zh: -->僵局*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含多个线程或可执行段，它们等待彼此释放必要的锁定，从而导致死锁。

### Description:

The software contains multiple threads or executable segments that are waiting for each other to release a necessary lock, resulting in deadlock.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 834

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Excessive Iteration zh: -->过度迭代*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件执行迭代或循环而不充分限制循环执行的次数。

### Description:

The software performs an iteration or loop without sufficiently limiting the number of times that the loop is executed.

### 详细描述:

如果迭代可能受到攻击者的影响，则此弱点可能允许攻击者消耗过多的资源，如CPU或内存。在许多情况下，循环不需要是无限的，以便导致足够的资源消耗，从而对软件或其主机系统产生不利影响;它取决于每次迭代消耗的资源量。

### Extended Description:

If the iteration can be influenced by an attacker, this weakness could allow attackers to consume excessive resources such as CPU or memory. In many cases, a loop does not need to be infinite in order to cause enough resource consumption to adversely affect the software or its host system; it depends on the amount of resources consumed per iteration.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 835

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Loop with Unreachable Exit Condition ('Infinite Loop') zh: -->循环带有无法到达的退出条件（'无限循环'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序包含具有无法到达的退出条件的迭代或循环，即无限循环。

### Description:

The program contains an iteration or loop with an exit condition that cannot be reached, i.e., an infinite loop.

### 详细描述:

如果循环可能受到攻击者的影响，则此弱点可能允许攻击者消耗过多的资源，如CPU或内存。

### Extended Description:

If the loop can be influenced by an attacker, this weakness could allow attackers to consume excessive resources such as CPU or memory.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 836

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of Password Hash Instead of Password for Authentication zh: -->使用密码哈希而不是密码进行身份验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在数据存储中记录密码哈希，从客户端接收密码哈希，并将提供的哈希与从数据存储获得的哈希进行比较。

### Description:

The software records password hashes in a data store, receives a hash of a password from a client, and compares the supplied hash to the hash obtained from the data store.

### 详细描述:

某些身份验证机制依赖于客户端生成密码的哈希值，可能会减少服务器上的负载或避免通过网络发送密码。但是，当客户端用于生成散列时，攻击者可以通过获取散列的副本来绕过身份验证，例如，通过使用SQL注入来破坏身份验证凭据的数据库，或利用信息泄露。然后，攻击者可以使用修改后的客户端重放被盗的哈希，而无需了解原始密码。  
因此，与客户端散列的服务器端比较不提供比使用没有散列的密码更多的安全性。

### Extended Description:

Some authentication mechanisms rely on the client to generate the hash for a password, possibly to reduce load on the server or avoid sending the password across the network. However, when the client is used to generate the hash, an attacker can bypass the authentication by obtaining a copy of the hash, e.g. by using SQL injection to compromise a database of authentication credentials, or by exploiting an information exposure. The attacker could then use a modified client to replay the stolen hash without having knowledge of the original password.  
As a result, the server-side comparison against a client-side hash does not provide any more security than the use of passwords without hashing.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 837

提交日期 I`m don`t know---> 修改日期 2012-05-11

* **Weakness Name:** *en: --> Improper Enforcement of a Single, Unique Action zh: -->单一，独特行动的不当执行*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件要求参与者应该只能执行一次操作，或者只能执行一个唯一操作，但软件不会强制执行或不正确地执行此限制。

### Description:

The software requires that an actor should only be able to perform an action once, or to have only one unique action, but the software does not enforce or improperly enforces this restriction.

### 详细描述:

在各种应用中，仅期望用户执行一次特定动作，例如投票，请求退款或进行购买。如果未强制执行此限制，有时可能会产生安全隐患。例如，在投票应用程序中，攻击者可以通过多次投票来尝试“填充投票箱”。如果这些投票分开计算，则攻击者可以直接影响谁赢得投票。根据软件的用途，这可能会产生重大的业务影响。

### Extended Description:

In various applications, a user is only expected to perform a certain action once, such as voting, requesting a refund, or making a purchase. When this restriction is not enforced, sometimes this can have security implications. For example, in a voting application, an attacker could attempt to "stuff the ballot box" by voting multiple times. If these votes are counted separately, then the attacker could directly affect who wins the vote. This could have significant business impact depending on the purpose of the software.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 838

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Inappropriate Encoding for Output Context zh: -->输出上下文的不适当编码*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

在向下游组件生成输出时，软件使用或指定编码，但指定的编码与下游组件预期的编码不同。

### Description:

The software uses or specifies an encoding when generating output to a downstream component, but the specified encoding is not the same as the encoding that is expected by the downstream component.

### 详细描述:

这种弱点可能导致下游组件使用解码方法，该方法产生的数据与软件要发送的数据不同。当使用错误的编码时 - 即使密切相关 - 下游组件可能会错误地解码数据。当控制和数据之间提供的边界被无意中断时，这会产生安全性后果，因为结果数据可能会引入控制字符或软件未发送的特殊元素。然后，结果数据可用于绕过保护机制，例如输入验证，并启用注入攻击。  
虽然使用输出编码对于确保组件之间的通信是准确的至关重要，但使用错误的编码 - 即使密切相关 - 可能会导致下游组件误解输出。  
例如，HTML实体编码用于网页的HTML主体中的元素。但是，程序员可能在生成输出时使用实体编码，该输出用于HTML标记的属性，该标记可能包含不受HTML编码影响的功能Javascript。  
虽然Web应用程序最受关注此问题，但这种弱点可能适用于使用可支持多种编码的通信流的任何类型的软件。

### Extended Description:

This weakness can cause the downstream component to use a decoding method that produces different data than what the software intended to send. When the wrong encoding is used - even if closely related - the downstream component could decode the data incorrectly. This can have security consequences when the provided boundaries between control and data are inadvertently broken, because the resulting data could introduce control characters or special elements that were not sent by the software. The resulting data could then be used to bypass protection mechanisms such as input validation, and enable injection attacks.  
While using output encoding is essential for ensuring that communications between components are accurate, the use of the wrong encoding - even if closely related - could cause the downstream component to misinterpret the output.  
For example, HTML entity encoding is used for elements in the HTML body of a web page. However, a programmer might use entity encoding when generating output for that is used within an attribute of an HTML tag, which could contain functional Javascript that is not affected by the HTML encoding.  
While web applications have received the most attention for this problem, this weakness could potentially apply to any type of software that uses a communications stream that could support multiple encodings.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 839

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Numeric Range Comparison Without Minimum Check zh: -->没有最小检查的数字范围比较*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

程序检查一个值以确保它小于或等于最大值，但它也不验证该值是否大于或等于最小值。

### Description:

The program checks a value to ensure that it is less than or equal to a maximum, but it does not also verify that the value is greater than or equal to the minimum.

### 详细描述:

某些程序使用有符号整数或浮点数，即使它们的值仅为正数或0时。输入验证检查可能会假定该值为正，并且仅检查最大值。如果值为负，但代码假定值为正，则可能产生错误。如果负值用于内存分配，阵列访问，缓冲区访问等，则该错误可能具有安全性后果。最终，该错误可能导致缓冲区溢出或其他类型的内存损坏。  
在仅有正面的上下文中使用负数可能会对其他类型的资源产生安全影响。例如，购物车可能会检查用户没有请求超过10个项目，但是对-3项目的请求可能会导致应用程序计算负价格并记入攻击者的帐户。

### Extended Description:

Some programs use signed integers or floats even when their values are only expected to be positive or 0. An input validation check might assume that the value is positive, and only check for the maximum value. If the value is negative, but the code assumes that the value is positive, this can produce an error. The error may have security consequences if the negative value is used for memory allocation, array access, buffer access, etc. Ultimately, the error could lead to a buffer overflow or other type of memory corruption.  
The use of a negative number in a positive-only context could have security implications for other types of resources. For example, a shopping cart might check that the user is not requesting more than 10 items, but a request for -3 items could cause the application to calculate a negative price and credit the attacker's account.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 841

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Enforcement of Behavioral Workflow zh: -->行为工作流程的不当执行*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件支持一个会话，其中一个actor必须执行多个行为，但它不能正确确保actor按所需顺序执行行为。

### Description:

The software supports a session in which more than one behavior must be performed by an actor, but it does not properly ensure that the actor performs the behaviors in the required sequence.

### 详细描述:

通过以意外顺序执行操作或通过省略步骤，攻击者可以操纵软件的业务逻辑或使其进入无效状态。在某些情况下，这也可能暴露出由此产生的弱点。  
例如，文件共享协议可能要求actor在能够传输文件之前执行单独的步骤以提供用户名，然后提供密码。如果文件共享服务器接受密码命令后跟传输命令，但未提供任何用户名，则软件可能仍会执行传输。  
请注意，这与CWE-696不同，后者侧重于软件以错误顺序执行操作的时间;这个条目密切相关，但它的重点是确保参与者以正确的顺序执行操作。  
与工作流程相关的行为包括：  
  
  
步骤按预期顺序执行。  
不会省略所需的步骤。  
步骤不会中断。  
步骤及时进行。

### Extended Description:

By performing actions in an unexpected order, or by omitting steps, an attacker could manipulate the business logic of the software or cause it to enter an invalid state. In some cases, this can also expose resultant weaknesses.  
For example, a file-sharing protocol might require that an actor perform separate steps to provide a username, then a password, before being able to transfer files. If the file-sharing server accepts a password command followed by a transfer command, without any username being provided, the software might still perform the transfer.  
Note that this is different than CWE-696, which focuses on when the software performs actions in the wrong sequence; this entry is closely related, but it is focused on ensuring that the actor performs actions in the correct sequence.  
Workflow-related behaviors include:  
  
  
Steps are performed in the expected order.  
Required steps are not omitted.  
Steps are not interrupted.  
Steps are performed in a timely fashion.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

这种弱点通常与业务逻辑缺陷有关，除非它产生由此产生的弱点。虽然业务漏洞的利用经常发生在现实世界的系统中，但业务逻辑缺陷的分类仍未得到充分研究，许多应用漏洞的研究人员对它们进行了调查。最重要的是Web应用程序。社区内部对于这些问题是否代表特别新的概念，或者它们是否是众所周知的原则的变化存在争议。许多业务逻辑缺陷似乎都面向业务流程，应用程序流和行为序列，这些缺陷在CWE中没有像输入验证，内存管理等相关的弱点那样得到充分体现。

Weakness ID: 842

提交日期 I`m don`t know---> 修改日期 2011-06-01

* **Weakness Name:** *en: --> Placement of User into Incorrect Group zh: -->将用户放置到不正确的组中*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

软件或管理员将用户置于不正确的组中。

### Description:

The software or the administrator places a user into an incorrect group.

### 详细描述:

如果不正确的组具有比预期组更多的访问权限或特权，则用户可能能够绕过预期的安全策略来访问意外资源或执行意外操作。访问控制系统可能无法检测到此组成员身份的恶意使用情况。

### Extended Description:

If the incorrect group has more access or privileges than the intended group, the user might be able to bypass intended security policy to access unexpected resources or perform unexpected actions. The access-control system might not be able to detect malicious usage of this group membership.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 843

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Access of Resource Using Incompatible Type ('Type Confusion') zh: -->使用不兼容类型访问资源（'类型混淆'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

程序使用一种类型分配或初始化诸如指针，对象或变量之类的资源，但稍后它使用与原始类型不兼容的类型访问该资源。

### Description:

The program allocates or initializes a resource such as a pointer, object, or variable using one type, but it later accesses that resource using a type that is incompatible with the original type.

### 详细描述:

当程序使用不兼容类型访问资源时，这可能会触发逻辑错误，因为资源没有预期的属性。在没有内存安全性的语言中，例如C和C ++，类型混淆可能导致越界内存访问。  
虽然在C中解析具有许多不同嵌入对象类型的数据时，这种弱点经常与联合相关联，但是它可以存在于可以以多种方式解释相同变量或存储器位置的任何应用程序中。  
这个弱点并不是C和C ++独有的。例如，可以通过在期望标量时提供数组参数来触发PHP应用程序中的错误，反之亦然。像Perl这样的语言，当它被访问时，它执行一种类型的变量的自动转换，就像它是另一种类型一样，也可以包含这些问题。

### Extended Description:

When the program accesses the resource using an incompatible type, this could trigger logical errors because the resource does not have expected properties. In languages without memory safety, such as C and C++, type confusion can lead to out-of-bounds memory access.  
While this weakness is frequently associated with unions when parsing data with many different embedded object types in C, it can be present in any application that can interpret the same variable or memory location in multiple ways.  
This weakness is not unique to C and C++. For example, errors in PHP applications can be triggered by providing array parameters when scalars are expected, or vice versa. Languages such as Perl, which perform automatic conversion of a variable of one type when it is accessed as if it were another type, can also contain these issues.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

任何类型不安全的编程语言都可能存在这种弱点。

Weakness ID: 862

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Missing Authorization zh: -->缺少授权*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当actor尝试访问资源或执行操作时，该软件不执行授权检查。

### Description:

The software does not perform an authorization check when an actor attempts to access a resource or perform an action.

### 详细描述:

假设具有给定身份的用户，授权是基于用户的权限和适用于资源的任何权限或其他访问控制规范来确定该用户是否可以访问给定资源的过程。  
如果未应用访问控制检查，则用户可以访问数据或执行不应允许执行的操作。这可能导致各种各样的问题，包括信息泄露，拒绝服务和任意代码执行。

### Extended Description:

Assuming a user with a given identity, authorization is the process of determining whether that user can access a given resource, based on the user's privileges and any permissions or other access-control specifications that apply to the resource.  
When access control checks are not applied, users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information exposures, denial of service, and arbitrary code execution.

### 问题背景 (Background Detail):

访问控制列表（ACL）表示对给定对象具有权限的人员/内容。不同的操作系统以不同的方式实现（ACL）。在UNIX中，有三种类型的权限：读取，写入和执行。用户分为三个类用于文件访问：所有者，组所有者以及每个类具有单独权限集的所有其他用户。在Windows NT中，文件的权限有四种基本类型：“无访问权限”，“读取权限”，“更改访问权限”和“完全控制权”。 Windows NT扩展了UNIX中三种类型用户的概念，包括用户和组列表及其相关权限。用户可以创建对象（文件）并为该对象分配指定的权限。

### 笔记 (Notes):

没有笔记

Weakness ID: 863

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Incorrect Authorization zh: -->授权不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

当actor尝试访问资源或执行操作时，软件会执行授权检查，但它不能正确执行检查。这允许攻击者绕过预期的访问限制。

### Description:

The software performs an authorization check when an actor attempts to access a resource or perform an action, but it does not correctly perform the check. This allows attackers to bypass intended access restrictions.

### 详细描述:

假设具有给定身份的用户，授权是基于用户的权限和适用于资源的任何权限或其他访问控制规范来确定该用户是否可以访问给定资源的过程。  
如果未正确应用访问控制检查，则用户可以访问数据或执行不应允许执行的操作。这可能导致各种各样的问题，包括信息泄露，拒绝服务和任意代码执行。

### Extended Description:

Assuming a user with a given identity, authorization is the process of determining whether that user can access a given resource, based on the user's privileges and any permissions or other access-control specifications that apply to the resource.  
When access control checks are incorrectly applied, users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information exposures, denial of service, and arbitrary code execution.

### 问题背景 (Background Detail):

访问控制列表（ACL）表示对给定对象具有权限的人员/内容。不同的操作系统以不同的方式实现（ACL）。在UNIX中，有三种类型的权限：读取，写入和执行。用户分为三个类用于文件访问：所有者，组所有者以及每个类具有单独权限集的所有其他用户。在Windows NT中，文件的权限有四种基本类型：“无访问权限”，“读取权限”，“更改访问权限”和“完全控制权”。 Windows NT扩展了UNIX中三种类型用户的概念，包括用户和组列表及其相关权限。用户可以创建对象（文件）并为该对象分配指定的权限。

### 笔记 (Notes):

没有笔记

Weakness ID: 908

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Uninitialized Resource zh: -->使用未初始化的资源*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用尚未正确初始化的资源。

### Description:

The software uses a resource that has not been properly initialized.

### 详细描述:

当关联资源预期具有某些属性或值时，这可能会产生安全隐患。

### Extended Description:

This can have security implications when the associated resource is expected to have certain properties or values.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 909

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Missing Initialization of Resource zh: -->缺少资源的初始化*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不初始化关键资源。

### Description:

The software does not initialize a critical resource.

### 详细描述:

许多资源需要初始化才能正确使用。如果资源未初始化，则可能包含不可预测或过期的数据，或者可能将其初始化为无效的默认值。当资源预期具有某些属性或值时，这会产生安全隐患。

### Extended Description:

Many resources require initialization before they can be properly used. If a resource is not initialized, it could contain unpredictable or expired data, or it could be initialized to defaults that are invalid. This can have security implications when the resource is expected to have certain properties or values.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 910

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Expired File Descriptor zh: -->使用过期文件描述符*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在关闭后使用或访问文件描述符。

### Description:

The software uses or accesses a file descriptor after it has been closed.

### 详细描述:

在释放特定文件或设备的文件描述符之后，可以重复使用它。代码可能无法写入原始文件，因为重用的文件描述符可能引用不同的文件或设备。

### Extended Description:

After a file descriptor for a particular file or device has been released, it can be reused. The code might not write to the original file, since the reused file descriptor might reference a different file or device.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 911

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Improper Update of Reference Count zh: -->引用计数更新不当*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用引用计数来管理资源，但它不会更新或错误地更新引用计数。

### Description:

The software uses a reference count to manage a resource, but it does not update or incorrectly updates the reference count.

### 详细描述:

在跟踪有多少对象包含对特定资源的引用时，例如在内存管理或垃圾回收中，可以使用引用计数。当引用计数达到零时，可以取消分配或重用资源，因为没有更多的对象使用它。如果引用计数意外地达到零，则资源可能会过早释放，即使它仍在使用中。如果所有对象不再使用该资源，但引用计数不为零，则可能永远不会释放该资源。

### Extended Description:

Reference counts can be used when tracking how many objects contain a reference to a particular resource, such as in memory management or garbage collection. When the reference count reaches zero, the resource can be de-allocated or reused because there are no more objects that use it. If the reference count accidentally reaches zero, then the resource might be released too soon, even though it is still in use. If all objects no longer use the resource, but the reference count is not zero, then the resource might not ever be released.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 912

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Hidden Functionality zh: -->隐藏的功能*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含未记录的功能，不是规范的一部分，并且无法通过对软件用户或管理员显而易见的界面或命令序列进行访问。

### Description:

The software contains functionality that is not documented, not part of the specification, and not accessible through an interface or command sequence that is obvious to the software's users or administrators.

### 详细描述:

隐藏的功能可以采取多种形式，例如故意恶意代码，包含游戏等无关功能的“复活节彩蛋”，减少维护或支持成本的开发人员友好型快捷方式，如硬编码帐户等。从安全角度来看，甚至当功能不是故意恶意或破坏时，它可能会增加软件的攻击面并暴露超出预期功能已经暴露的其他弱点。即使不容易访问，隐藏的功能也可能对修改应用程序控制流的攻击很有用。

### Extended Description:

Hidden functionality can take many forms, such as intentionally malicious code, "Easter Eggs" that contain extraneous functionality such as games, developer-friendly shortcuts that reduce maintenance or support costs such as hard-coded accounts, etc. From a security perspective, even when the functionality is not intentionally malicious or damaging, it can increase the software's attack surface and expose additional weaknesses beyond what is already exposed by the intended functionality. Even if it is not easily accessible, the hidden functionality could be useful for attacks that modify the control flow of the application.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 913

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Control of Dynamically-Managed Code Resources zh: -->动态管理代码资源的不正确控制*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不能正确地限制对动态管理的代码资源（如变量，对象，类，属性，函数或可执行指令或语句）的读取或写入。

### Description:

The software does not properly restrict reading from or writing to dynamically-managed code resources such as variables, objects, classes, attributes, functions, or executable instructions or statements.

### 详细描述:

许多语言提供了强大的功能，允许程序员动态创建或修改现有代码或代码使用的资源，如变量和对象。虽然这些功能可以提供显着的灵活性并缩短开发时间，但如果攻击者能够以意想不到的方式直接影响这些代码资源，则会非常危险。

### Extended Description:

Many languages offer powerful features that allow the programmer to dynamically create or modify existing code, or resources used by code such as variables and objects. While these features can offer significant flexibility and reduce development time, they can be extremely dangerous if attackers can directly influence these code resources in unexpected ways.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 914

提交日期 I`m don`t know---> 修改日期 2017-01-19

* **Weakness Name:** *en: --> Improper Control of Dynamically-Identified Variables zh: -->动态识别变量的不正确控制*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件不能正确地限制对动态识别变量的读取或写入。

### Description:

The software does not properly restrict reading from or writing to dynamically-identified variables.

### 详细描述:

许多语言都提供了强大的功能，允许程序员访问由输入字符串指定的任意变量。虽然这些功能可以提供显着的灵活性并缩短开发时间，但如果攻击者可以修改具有安全隐患的非预期变量，则这些功能非常危险。

### Extended Description:

Many languages offer powerful features that allow the programmer to access arbitrary variables that are specified by an input string. While these features can offer significant flexibility and reduce development time, they can be extremely dangerous if attackers can modify unintended variables that have security implications.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 915

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improperly Controlled Modification of Dynamically-Determined Object Attributes zh: -->动态确定的对象属性的不正确控制修改*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件接收来自上游组件的输入，该组件指定要在对象中初始化或更新的多个属性，属性或字段，但是它不能正确地控制可以修改哪些属性。

### Description:

The software receives input from an upstream component that specifies multiple attributes, properties, or fields that are to be initialized or updated in an object, but it does not properly control which attributes can be modified.

### 详细描述:

如果对象包含仅供内部使用的属性，则其意外修改可能会导致漏洞。  
这种弱点有时可以通过特定于语言的机制来实现，例如质量分配，自动绑定或对象注入。

### Extended Description:

If the object contains attributes that were only intended for internal use, then their unexpected modification could lead to a vulnerability.  
This weakness is sometimes known by the language-specific mechanisms that make it possible, such as mass assignment, autobinding, or object injection.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

CWE-502和CWE-915之间的关系需要进一步探索。 CWE-915的范围更窄，适用于对象修改，不一定用于反序列化。

Weakness ID: 916

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Use of Password Hash With Insufficient Computational Effort zh: -->使用密码哈希计算努力不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件为密码生成哈希，但它使用的方案不能提供足够的计算量，使密码破解攻击不可行或昂贵。

### Description:

The software generates a hash for a password, but it uses a scheme that does not provide a sufficient level of computational effort that would make password cracking attacks infeasible or expensive.

### 详细描述:

许多密码存储机制计算散列并存储散列，而不是将原始密码存储在明文中。在此设计中，身份验证涉及接受传入密码，计算其哈希值，并将其与存储的哈希值进行比较。  
许多哈希算法被设计为以最小的开销快速执行，甚至是加密哈希。但是，这种效率是密码存储的一个问题，因为它可以减少攻击者的暴力密码破解工作量。如果攻击者可以通过其他方法（例如存储哈希值的数据库上的SQL注入）获取哈希值，则攻击者可以离线存储哈希值，并使用各种技术通过有效计算哈希值来破解密码。如果没有内置工作负载，现代攻击可以在很短的时间内使用大规模并行计算（如云计算）和GPU，ASIC计算大量哈希值，甚至耗尽所有可能密码的整个空间或FPGA硬件。在这种情况下，有效的哈希算法可以帮助攻击者。  
散列方案的若干属性与其针对离线，大规模并行攻击的强度相关：  
  
  
计算哈希值所需的CPU时间量（“拉伸”）  
计算哈希所需的内存量（“内存难”操作）  
包括随机值和密码，作为哈希计算的输入（“salting”）  
给定散列，除了通过猜测可能的输入（“单向”散列）之外，没有已知的方法来确定产生该散列值的输入（例如，密码）。  
相对于该方案可以生成的所有可能哈希的数量，为多个不同输入产生相同哈希的可能性很小（“抗冲突性”）  
  
  
请注意，软件的安全要求可能因环境和密码值而异。不同的方案可能无法提供所有这些属性，但仍可为环境提供足够的安全性。相反，解决方案在保留一个属性方面可能非常强大，对于另一个属性的攻击仍然非常弱，或者它可能无法显着降低大规模并行攻击的效率。

### Extended Description:

Many password storage mechanisms compute a hash and store the hash, instead of storing the original password in plaintext. In this design, authentication involves accepting an incoming password, computing its hash, and comparing it to the stored hash.  
Many hash algorithms are designed to execute quickly with minimal overhead, even cryptographic hashes. However, this efficiency is a problem for password storage, because it can reduce an attacker's workload for brute-force password cracking. If an attacker can obtain the hashes through some other method (such as SQL injection on a database that stores hashes), then the attacker can store the hashes offline and use various techniques to crack the passwords by computing hashes efficiently. Without a built-in workload, modern attacks can compute large numbers of hashes, or even exhaust the entire space of all possible passwords, within a very short amount of time, using massively-parallel computing (such as cloud computing) and GPU, ASIC, or FPGA hardware. In such a scenario, an efficient hash algorithm helps the attacker.  
There are several properties of a hash scheme that are relevant to its strength against an offline, massively-parallel attack:  
  
  
The amount of CPU time required to compute the hash ("stretching")  
The amount of memory required to compute the hash ("memory-hard" operations)  
Including a random value, along with the password, as input to the hash computation ("salting")  
Given a hash, there is no known way of determining an input (e.g., a password) that produces this hash value, other than by guessing possible inputs ("one-way" hashing)  
Relative to the number of all possible hashes that can be generated by the scheme, there is a low likelihood of producing the same hash for multiple different inputs ("collision resistance")  
  
  
Note that the security requirements for the software may vary depending on the environment and the value of the passwords. Different schemes might not provide all of these properties, yet may still provide sufficient security for the environment. Conversely, a solution might be very strong in preserving one property, which still being very weak for an attack against another property, or it might not be able to significantly reduce the efficiency of a massively-parallel attack.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 917

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Neutralization of Special Elements used in an Expression Language Statement ('Expression Language Injection') zh: -->表达语言语句中使用的特殊元素的不正当中和（'表达式语言注入'）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用来自上游组件的外部影响输入在Java Server Page（JSP）中构造表达式语言（EL）语句的全部或部分，但它不会中和或不正确地中和可能之前修改预期EL语句的特殊元素它被执行了。

### Description:

The software constructs all or part of an expression language (EL) statement in a Java Server Page (JSP) using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended EL statement before it is executed.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

在Spring 3.0.5及更早版本的某些版本中，存在一个漏洞（CVE-2011-2730），其中表达式语言标记将被评估两次，从而有效地将任何应用程序暴露给EL注入。但是，即使对于更高版本，依赖于配置，仍然存在这种弱点。

Weakness ID: 918

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Server-Side Request Forgery (SSRF) zh: -->服务器端请求伪造（SSRF）*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web服务器从上游组件接收URL或类似请求并检索此URL的内容，但它不足以确保将请求发送到预期目标。

### Description:

The web server receives a URL or similar request from an upstream component and retrieves the contents of this URL, but it does not sufficiently ensure that the request is being sent to the expected destination.

### 详细描述:

通过向意外的主机或端口提供URL，攻击者可以看到服务器正在发送请求，可能绕过阻止攻击者直接访问URL的防火墙等访问控制。服务器可以用作代理来对内部网络中的主机进行端口扫描，使用其他URL，例如可以访问系统上的文档（使用file：//），或使用其他协议，如gopher：//或tftp ：//，可以提供对请求内容的更大控制。

### Extended Description:

By providing URLs to unexpected hosts or ports, attackers can make it appear that the server is sending the request, possibly bypassing access controls such as firewalls that prevent the attackers from accessing the URLs directly. The server can be used as a proxy to conduct port scanning of hosts in internal networks, use other URLs such as that can access documents on the system (using file://), or use other protocols such as gopher:// or tftp://, which may provide greater control over the contents of requests.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

CWE-918（SSRF）和CWE-611（XXE）密切相关，因为它们都涉及与Web相关的技术，并且可以向意外目的地发出出站请求。但是，XXE可以在客户端执行，或者在软件不直接作为服务器的其他环境中执行，因此SSRF首字母缩略词的“服务器”部分不一定适用。

Weakness ID: 920

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Improper Restriction of Power Consumption zh: -->不正确的功耗限制*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在电源是有限资源的环境中运行，该资源不能自动补充，但软件不能正确地限制其操作消耗的电量。

### Description:

The software operates in an environment in which power is a limited resource that cannot be automatically replenished, but the software does not properly restrict the amount of power that its operation consumes.

### 详细描述:

在诸如嵌入式或移动设备之类的环境中，电源可以是有限的资源，例如电池，其不能由软件本身自动补充，并且设备可能不总是直接连接到可靠的电源。如果软件过快地使用太多电力，则可能导致设备（以及随后的软件）在电源恢复之前停止工作，或者由于电力成本增加而增加设备所有者的财务负担。  
应用程序的正常运行会消耗功率。但是，在某些情况下，攻击者可能会使应用程序消耗的功率超出预期，使用以下组件：  
  
  
显示  
中央处理器  
磁盘I / O.  
全球定位系统  
声音  
麦克风  
USB接口

### Extended Description:

In environments such as embedded or mobile devices, power can be a limited resource such as a battery, which cannot be automatically replenished by the software itself, and the device might not always be directly attached to a reliable power source. If the software uses too much power too quickly, then this could cause the device (and subsequently, the software) to stop functioning until power is restored, or increase the financial burden on the device owner because of increased power costs.  
Normal operation of an application will consume power. However, in some cases, an attacker could cause the application to consume more power than intended, using components such as:  
  
  
Display  
CPU  
Disk I/O  
GPS  
Sound  
Microphone  
USB interface

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 921

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Storage of Sensitive Data in a Mechanism without Access Control zh: -->在没有访问控制的机制中存储敏感数据*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件将敏感信息存储在没有内置访问控制的文件系统或设备中。

### Description:

The software stores sensitive information in a file system or device that does not have built-in access control.

### 详细描述:

虽然许多现代文件系统或设备使用某种形式的访问控制来限制对数据的访问，但并非所有存储机制都具有此功能。例如，存储卡，软盘，CD和USB设备通常可供系统内的任何用户访问。当敏感数据存储在多用户环境中的这些机制中时，这可能成为问题，因为系统上的任何人都可以读取或写入此数据。  
在Android设备上，外部存储通常可由设备上的其他应用程序全局读取和写入。外部存储也可以通过移动设备的USB连接轻松访问，也可以通过设备的存储卡端口进行物理访问。

### Extended Description:

While many modern file systems or devices utilize some form of access control in order to restrict access to data, not all storage mechanisms have this capability. For example, memory cards, floppy disks, CDs, and USB devices are typically made accessible to any user within the system. This can become a problem when sensitive data is stored in these mechanisms in a multi-user environment, because anybody on the system can read or write this data.  
On Android devices, external storage is typically globally readable and writable by other applications on the device. External storage may also be easily accessible through the mobile device's USB connection or physically accessible through the device's memory card port.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 922

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Insecure Storage of Sensitive Information zh: -->敏感信息的不安全存储*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件存储敏感信息，而不适当地限制未经授权的演员的读或写访问。

### Description:

The software stores sensitive information without properly limiting read or write access by unauthorized actors.

### 详细描述:

如果未正确限制读取访问权限，则攻击者可以窃取敏感信息。如果写访问权限未得到适当限制，则攻击者可以修改并可能删除数据，从而导致错误的结果和可能的拒绝服务。

### Extended Description:

If read access is not properly restricted, then attackers can steal the sensitive information. If write access is not properly restricted, then attackers can modify and possibly delete the data, causing incorrect results and possibly a denial of service.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

敏感信息的不安全存储（CWE-922）与敏感信息的缺失加密（CWE-311）之间存在重叠关系。加密通常用于防止攻击者读取敏感数据。但是，加密不会阻止攻击者擦除或覆盖数据。

Weakness ID: 923

提交日期 2014-02-18---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Restriction of Communication Channel to Intended Endpoints zh: -->通信信道对预期端点的不正确限制*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件为特权或受保护的操作建立到（或来自）端点的通信通道，但是它没有正确地确保它与正确的端点通信。

### Description:

The software establishes a communication channel to (or from) an endpoint for privileged or protected operations, but it does not properly ensure that it is communicating with the correct endpoint.

### 详细描述:

攻击者可能能够欺骗来自不同系统或进程的预期端点，从而获得与预期端点相同的访问级别。  
虽然此问题经常涉及基于网络的客户端和服务器之间的身份验证，但其他类型的通信通道和端点可能存在此缺陷。

### Extended Description:

Attackers might be able to spoof the intended endpoint from a different system or process, thus gaining the same level of access as the intended endpoint.  
While this issue frequently involves authentication between network-based clients and servers, other types of communication channels and endpoints can have this weakness.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目将在以后的CWE版本中更加全面。

Weakness ID: 924

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Enforcement of Message Integrity During Transmission in a Communication Channel zh: -->通信信道传输过程中消息完整性的不正确执行*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件与端点建立通信信道并从该端点接收消息，但是它不能充分确保在传输期间不修改消息。

### Description:

The software establishes a communication channel with an endpoint and receives a message from that endpoint, but it does not sufficiently ensure that the message was not modified during transmission.

### 详细描述:

中间人（MITM）攻击者可能能够修改消息并欺骗端点。

### Extended Description:

A man-in-the-middle (MITM) attacker might be able to modify the message and spoof the endpoint.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目将在以后的CWE版本中更加全面。

Weakness ID: 925

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Improper Verification of Intent by Broadcast Receiver zh: -->广播接收机对意图的不正确验证*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Android应用程序使用广播接收器接收Intent但未正确验证Intent来自授权来源。

### Description:

The Android application uses a Broadcast Receiver that receives an Intent but does not properly verify that the Intent came from an authorized source.

### 详细描述:

由操作字符串标识的某些类型的Intent只能由操作系统本身广播，而不能由第三方应用程序广播。但是，当应用程序注册接收这些隐式系统意图时，它也会注册以接收任何显式意图。虽然恶意应用程序无法发送隐式系统意图，但它可以向目标应用程序发送显式意图，这可能假设任何接收到的意图是有效的隐式系统意图而不是来自其他应用程序的明确意图。这可能会导致意外行为。

### Extended Description:

Certain types of Intents, identified by action string, can only be broadcast by the operating system itself, not by third-party applications. However, when an application registers to receive these implicit system intents, it is also registered to receive any explicit intents. While a malicious application cannot send an implicit system intent, it can send an explicit intent to the target application, which may assume that any received intent is a valid implicit system intent and not an explicit intent from another application. This may lead to unintended behavior.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目将在以后的CWE版本中更加全面。

Weakness ID: 926

提交日期 2014-02-18---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Export of Android Application Components zh: -->Android应用程序组件的不正确导出*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Android应用程序导出一个供其他应用程序使用的组件，但不能正确限制哪些应用程序可以启动组件或访问它包含的数据。

### Description:

The Android application exports a component for use by other applications, but does not properly restrict which applications can launch the component or access the data it contains.

### 详细描述:

不正确地导出组件的攻击和后果可能取决于导出的组件：  
  
  
如果不限制对导出的活动的访问，则任何应用程序都可以启动该活动。这可能允许恶意应用程序获得对敏感信息的访问，修改应用程序的内部状态，或欺骗用户与受害者应用程序交互，同时相信他们仍在与恶意应用程序交互。  
如果不限制对导出的服务的访问，则任何应用程序都可以启动并绑定到服务。根据公开的功能，这可能允许恶意应用程序执行未经授权的操作，获取对敏感信息的访问权限或破坏应用程序的内部状态。  
如果对Content Provider的访问不仅限于预期的应用程序，则恶意应用程序可能能够访问敏感数据。请注意，在4.2之前的Android中，内容提供程序会自动导出，除非它已显式声明为NOT导出。

### Extended Description:

The attacks and consequences of improperly exporting a component may depend on the exported component:  
  
  
If access to an exported Activity is not restricted, any application will be able to launch the activity. This may allow a malicious application to gain access to sensitive information, modify the internal state of the application, or trick a user into interacting with the victim application while believing they are still interacting with the malicious application.  
If access to an exported Service is not restricted, any application may start and bind to the Service. Depending on the exposed functionality, this may allow a malicious application to perform unauthorized actions, gain access to sensitive information, or corrupt the internal state of the application.  
If access to a Content Provider is not restricted to only the expected applications, then malicious applications might be able to access the sensitive data. Note that in Android before 4.2, the Content Provider is automatically exported unless it has been explicitly declared as NOT exported.

### 问题背景 (Background Detail):

可以在Android应用程序中导出三种类型的组件。活动活动是一个应用程序组件，它为用户提供与之交互的UI。典型的应用程序将具有多个执行不同功能的活动屏幕，例如主活动屏幕和单独的设置活动屏幕。 Service A Service是一个应用程序组件，由另一个组件启动以在后台执行操作，即使在调用组件终止后也是如此。服务没有用户可见的UI组件。内容提供程序内容提供程序机制可用于与其他应用程序共享数据或在同一应用程序内部共享数据。

### 笔记 (Notes):

没有笔记

Weakness ID: 927

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Use of Implicit Intent for Sensitive Communication zh: -->使用隐含意图进行敏感交流*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Android应用程序使用隐式意图将敏感数据传输到其他应用程序。

### Description:

The Android application uses an implicit intent for transmitting sensitive data to other applications.

### 详细描述:

由于隐式intent没有指定接收数据的特定应用程序，因此任何应用程序都可以通过对该意图使用Intent Filter来处理intent。这可以允许不受信任的应用程序获取敏感数据。标准广播意图有两种变体，有序和粘性。  
有序广播意图按接收者声明的优先级顺序发送给一系列注册接收者。恶意接收器可以通过阻止广播在链中进一步传播而使自己具有高优先级并导致拒绝服务。还存在恶意数据修改的可能性，因为接收器也可能在将其传递给下一个接收器之前改变Intent内的数据。下游组件无法断言数据在链中的早期未被更改。  
在最初的广播之后仍然可以访问粘性广播意图。旧的粘性意图将再次播放给将来注册它的任何新接收器，这大大增加了信息暴露的可能性。此外，粘性广播不能受可能适用于其他类型意图的权限的保护。  
此外，任何广播意图可以包括引用接收组件通常不具有访问权限的数据的URI。意图的发送者可以包括特权，授予接收者对意图中包含的特定URI的读或写访问权。拦截此意图的恶意接收器也将获得这些权限，并能够在指定的URI处读取或写入资源。

### Extended Description:

Since an implicit intent does not specify a particular application to receive the data, any application can process the intent by using an Intent Filter for that intent. This can allow untrusted applications to obtain sensitive data. There are two variations on the standard broadcast intent, ordered and sticky.  
Ordered broadcast intents are delivered to a series of registered receivers in order of priority as declared by the Receivers. A malicious receiver can give itself a high priority and cause a denial of service by stopping the broadcast from propagating further down the chain. There is also the possibility of malicious data modification, as a receiver may also alter the data within the Intent before passing it on to the next receiver. The downstream components have no way of asserting that the data has not been altered earlier in the chain.  
Sticky broadcast intents remain accessible after the initial broadcast. An old sticky intent will be broadcast again to any new receivers that register for it in the future, greatly increasing the chances of information exposure over time. Also, sticky broadcasts cannot be protected by permissions that may apply to other kinds of intents.  
In addition, any broadcast intent may include a URI that references data that the receiving component does not normally have the privileges to access. The sender of the intent can include special privileges that grant the receiver read or write access to the specific URI included in the intent. A malicious receiver that intercepts this intent will also gain those privileges and be able to read or write the resource at the specified URI.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目将在以后的CWE版本中更加全面。

Weakness ID: 939

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Authorization in Handler for Custom URL Scheme zh: -->自定义URL方案处理程序中的不正确授权*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用自定义URL方案的处理程序，但它没有正确地限制哪些actor可以使用该方案调用处理程序。

### Description:

The software uses a handler for a custom URL scheme, but it does not properly restrict which actors can invoke the handler using the scheme.

### 详细描述:

移动平台和其他体系结构允许使用自定义URL方案来促进应用程序之间的通信。在iOS的情况下，这是进行应用程序间通信的唯一方法。该实现由开发人员自行决定，可能会在应用程序中打开安全漏洞。一个示例可能是潜在的危险功能，例如通过自定义URL方案修改文件。

### Extended Description:

Mobile platforms and other architectures allow the use of custom URL schemes to facilitate communication between applications. In the case of iOS, this is the only method to do inter-application communication. The implementation is at the developer's discretion which may open security flaws in the application. An example could be potentially dangerous functionality such as modifying files through a custom URL scheme.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 940

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Improper Verification of Source of a Communication Channel zh: -->通信渠道来源的不正确验证*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件建立一个通信通道来处理由一个actor发起的传入请求，但是它没有正确地验证该请求来自预期的来源。

### Description:

The software establishes a communication channel to handle an incoming request that has been initiated by an actor, but it does not properly verify that the request is coming from the expected origin.

### 详细描述:

当攻击者可以成功建立来自不受信任来源的通信通道时，攻击者可能能够获得权限并访问意外功能。

### Extended Description:

When an attacker can successfully establish a communication channel from an untrusted origin, the attacker may be able to gain privileges and access unexpected functionality.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

虽然许多访问控制问题涉及验证用户，但这个弱点更多的是验证通信信道本身的实际来源;在这种情况下可能没有任何“用户”。

Weakness ID: 941

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Incorrectly Specified Destination in a Communication Channel zh: -->通信通道中错误指定的目的地*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件创建一个通信通道来启动对actor的传出请求，但它没有正确指定该actor的预期目标。

### Description:

The software creates a communication channel to initiate an outgoing request to an actor, but it does not correctly specify the intended destination for that actor.

### 详细描述:

目的地的攻击者可能会欺骗受信任的服务器来窃取数据或导致拒绝服务。  
至少有两个明显的弱点可能导致软件与非预期目的地进行通信：  
  
  
如果软件允许攻击者控制指定的目标，则攻击者可以使其连接到不受信任或恶意的目标。例如，由于UDP是无连接协议，因此可以通过在数据包中指定错误的源地址来欺骗UDP数据包;当服务器接收到数据包并发送回复时，它将通过使用传入数据包的源 - 即错误源来指定目的地。然后可以欺骗服务器将流量发送到错误的主机，这对于隐藏真实的攻击源和进行分布式拒绝服务（DDoS）是有效的。作为另一个示例，服务器端请求伪造（SSRF）和XML外部实体（XXE）可用于欺骗服务器向由于防火墙限制而无法由攻击者直接访问的主机发出传出请求。  
如果软件错误地指定了目标，则可以控制此目标的攻击者可能会欺骗受信任的服务器。虽然最常见的情况可能是由于管理员的配置错误，但这可能是其他弱点造成的。例如，软件可能会错误地解析电子邮件或IP地址，并将敏感数据发送到非预期的目的地。作为另一个示例，Android应用程序可以使用“粘性广播”来与特定应用程序的接收器通信，但是由于粘性广播可以由\*任何\*接收器处理，这可以允许恶意应用程序访问仅受限制的数据。用于不同的应用程序。

### Extended Description:

Attackers at the destination may be able to spoof trusted servers to steal data or cause a denial of service.  
There are at least two distinct weaknesses that can cause the software to communicate with an unintended destination:  
  
  
If the software allows an attacker to control which destination is specified, then the attacker can cause it to connect to an untrusted or malicious destination. For example, because UDP is a connectionless protocol, UDP packets can be spoofed by specifying a false source address in the packet; when the server receives the packet and sends a reply, it will specify a destination by using the source of the incoming packet - i.e., the false source. The server can then be tricked into sending traffic to the wrong host, which is effective for hiding the real source of an attack and for conducting a distributed denial of service (DDoS). As another example, server-side request forgery (SSRF) and XML External Entity (XXE) can be used to trick a server into making outgoing requests to hosts that cannot be directly accessed by the attacker due to firewall restrictions.  
If the software incorrectly specifies the destination, then an attacker who can control this destination might be able to spoof trusted servers. While the most common occurrence is likely due to misconfiguration by an administrator, this can be resultant from other weaknesses. For example, the software might incorrectly parse an e-mail or IP address and send sensitive data to an unintended destination. As another example, an Android application may use a "sticky broadcast" to communicate with a receiver for a particular application, but since sticky broadcasts can be processed by \*any\* receiver, this can allow a malicious application to access restricted data that was only intended for a different application.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

此条目将在以后的CWE版本中更加全面。

Weakness ID: 942

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Overly Permissive Cross-domain Whitelist zh: -->过度宽容的跨域白名单*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用跨域策略文件，该文件包含不应信任的域。

### Description:

The software uses a cross-domain policy file that includes domains that should not be trusted.

### 详细描述:

跨域策略文件（Flash中的“crossdomain.xml”和Silverlight中的“clientaccesspolicy.xml”）定义域的白名单，允许服务器从中生成跨域请求。在发出跨域请求时，Flash或Silverlight客户端将首先在目标服务器上查找策略文件。如果找到，并且明确允许托管应用程序的域发出请求，则发出请求。  
因此，如果跨域策略文件包含不应受信任的域（例如使用通配符时），则应用程序可能会受到这些不受信任的域的攻击。  
过度宽松的策略文件允许在跨站点脚本（CWE-79）中看到许多相同的攻击。一旦用户执行了恶意Flash或Silverlight应用程序，他们就容易受到各种攻击。攻击者可以将受害者机器上的私人信息（例如可能包含会话信息的cookie）传输给攻击者。攻击者可以代表受害者向网站发送恶意请求，如果受害者具有管理该网站的管理员权限，则可能对该网站特别危险。  
在许多情况下，攻击可以在受害者甚至没有意识到的情况下发起。

### Extended Description:

A cross-domain policy file ("crossdomain.xml" in Flash and "clientaccesspolicy.xml" in Silverlight) defines a whitelist of domains from which a server is allowed to make cross-domain requests. When making a cross-domain request, the Flash or Silverlight client will first look for the policy file on the target server. If it is found, and the domain hosting the application is explicitly allowed to make requests, the request is made.  
Therefore, if a cross-domain policy file includes domains that should not be trusted, such as when using wildcards, then the application could be attacked by these untrusted domains.  
An overly permissive policy file allows many of the same attacks seen in Cross-Site Scripting (CWE-79). Once the user has executed a malicious Flash or Silverlight application, they are vulnerable to a variety of attacks. The attacker could transfer private information, such as cookies that may include session information, from the victim's machine to the attacker. The attacker could send malicious requests to a web site on behalf of the victim, which could be especially dangerous to the site if the victim has administrator privileges to manage that site.  
In many cases, the attack can be launched without the victim even being aware of it.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 943

提交日期 I`m don`t know---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Improper Neutralization of Special Elements in Data Query Logic zh: -->数据查询逻辑中特殊元素的中和不正确*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序生成一个查询，用于访问或操作数据库（如数据库）中的数据，但它不会中和或错误地中和可以修改查询的预期逻辑的特殊元素。

### Description:

The application generates a query intended to access or manipulate data in a data store such as a database, but it does not neutralize or incorrectly neutralizes special elements that can modify the intended logic of the query.

### 详细描述:

根据查询语言的功能，攻击者可以在查询中注入额外的逻辑，以便：  
  
  
修改预期的选择标准，从而改变返回，修改或以其他方式操纵哪些数据实体（例如，记录）  
在查询中附加其他命令  
返回比预期更多的实体  
返回的实体少于预期  
导致实体以意外方式排序  
  
  
执行其他命令或更改返回哪些实体的能力具有明显的风险。但是，当应用程序逻辑依赖于实体的顺序或数量时，这也可能导致漏洞。例如，如果应用程序查询期望仅返回指定管理用户的一个实体，但攻击者可以更改返回的实体，则可能导致逻辑返回常规用户的信息并错误地认为用户具有管理权限。  
虽然这种弱点最常与SQL注入相关联，但还有许多其他查询语言也受到注入攻击，包括HTSQL，LDAP，DQL，XQuery，Xpath和“NoSQL”语言。

### Extended Description:

Depending on the capabilities of the query language, an attacker could inject additional logic into the query to:  
  
  
Modify the intended selection criteria, thus changing which data entities (e.g., records) are returned, modified, or otherwise manipulated  
Append additional commands to the query  
Return more entities than intended  
Return fewer entities than intended  
Cause entities to be sorted in an unexpected way  
  
  
The ability to execute additional commands or change which entities are returned has obvious risks. But when the application logic depends on the order or number of entities, this can also lead to vulnerabilities. For example, if the application query expects to return only one entity that specifies an administrative user, but an attacker can change which entities are returned, this could cause the logic to return information for a regular user and incorrectly assume that the user has administrative privileges.  
While this weakness is most commonly associated with SQL injection, there are many other query languages that are also subject to injection attacks, including HTSQL, LDAP, DQL, XQuery, Xpath, and "NoSQL" languages.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

可以说，数据查询语言实际上是一种命令语言 - 尽管命令集有限 - 因此任何查询语言注入问题都可以被视为CWE-74的子代。但是，CWE-943旨在更好地组织面向查询的问题，将它们与功能完备的编程语言分开，并为没有自己的CWE标识符的许多查询语言提供更精确的标识符。

Weakness ID: 1004

提交日期 I`m don`t know---> 修改日期 2017-11-08

* **Weakness Name:** *en: --> Sensitive Cookie Without 'HttpOnly' Flag zh: -->没有'HttpOnly'标志的敏感Cookie*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用cookie来存储敏感信息，但cookie没有标记HttpOnly标志。

### Description:

The software uses a cookie to store sensitive information, but the cookie is not marked with the HttpOnly flag.

### 详细描述:

HttpOnly标志指示兼容的浏览器阻止客户端脚本访问cookie。在Set-Cookie HTTP响应头中包含HttpOnly标志有助于降低与跨站点脚本（XSS）相关的风险，其中攻击者的脚本代码可能会尝试读取cookie的内容并泄露获得的信息。设置后，支持该标志的浏览器不会通过XSS执行的客户端脚本向第三方显示cookie的内容。

### Extended Description:

The HttpOnly flag directs compatible browsers to prevent client-side script from accessing cookies. Including the HttpOnly flag in the Set-Cookie HTTP response header helps mitigate the risk associated with Cross-Site Scripting (XSS) where an attacker's script code might attempt to read the contents of a cookie and exfiltrate information obtained. When set, browsers that support the flag will not reveal the contents of the cookie to a third party via client-side script executed via XSS.

### 问题背景 (Background Detail):

HTTP cookie是属于特定网站的一小段数据，由用户的Web浏览器存储在用户的计算机上。可以将这些数据用于各种目的，包括将输入到表单字段中的信息保存，记录用户活动以及用于身份验证。用于保存或记录用户生成的信息的cookie由嵌入在网页中的脚本代码访问和修改。用于身份验证的Cookie由网站的服务器创建，并发送给用户以附加到将来的请求中。这些身份验证cookie通常不会被发送给用户的网页访问，而是应该附加到将来验证身份验证详细信息的请求。

### 笔记 (Notes):

没有笔记

Weakness ID: 1007

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Insufficient Visual Distinction of Homoglyphs Presented to User zh: -->提供给用户的同形视觉视觉区别不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件向用户显示信息或标识符，但是显示机制不容易使用户区分视觉上相似或相同的字形（同形字），这可能导致用户误解字形并执行意外的，不安全的动作。 。

### Description:

The software displays information or identifiers to a user, but the display mechanism does not make it easy for the user to distinguish between visually similar or identical glyphs (homoglyphs), which may cause the user to misinterpret a glyph and perform an unintended, insecure action.

### 详细描述:

某些字形，图片或图标在语义上可以与程序不同，而与人类用户看起来非常相似或相同。这些被称为同形体。例如，小写的“l”（椭圆形）和大写的“I”（眼睛）具有不同的字符代码，但是这些字符可以以完全相同的方式显示给用户，具体取决于字体。这也可以在不同的字符集之间发生。例如，拉丁大写字母“A”和希腊大写字母“Α”（Alpha）被程序视为不同，但可以以完全相同的方式显示给用户。重音标记也可能导致字母看起来非常相似，例如拉丁语大写字母标记“À”和等效的“À”具有尖锐的重音。  
攻击者可以利用这种视觉相似性来攻击诸如网络钓鱼等攻击。通过提供链接到攻击者控制的主机名，该主机名看起来像受害者信任的主机名。在对同形体的不同使用中，对手可以创建后门用户名，其在视觉上类似于常规用户的用户名，这使得系统管理员在查看日志时更难以检测恶意用户名。

### Extended Description:

Some glyphs, pictures, or icons can be semantically distinct to a program, while appearing very similar or identical to a human user. These are referred to as homoglyphs. For example, the lowercase "l" (ell) and uppercase "I" (eye) have different character codes, but these characters can be displayed in exactly the same way to a user, depending on the font. This can also occur between different character sets. For example, the Latin capital letter "A" and the Greek capital letter "Α" (Alpha) are treated as distinct by programs, but may be displayed in exactly the same way to a user. Accent marks may also cause letters to appear very similar, such as the Latin capital letter grave mark "À" and its equivalent "À" with the acute accent.  
Adversaries can exploit this visual similarity for attacks such as phishing, e.g. by providing a link to an attacker-controlled hostname that looks like a hostname that the victim trusts. In a different use of homoglyphs, an adversary may create a back door username that is visually similar to the username of a regular user, which then makes it more difficult for a system administrator to detect the malicious username while reviewing logs.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1021

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Improper Restriction of Rendered UI Layers or Frames zh: -->渲染的UI图层或框架的不正确限制*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序不限制或不正确地限制属于另一个应用程序或域的框架对象或UI层，这可能导致用户混淆用户正在与哪个界面进行交互。

### Description:

The web application does not restrict or incorrectly restricts frame objects or UI layers that belong to another application or domain, which can lead to user confusion about which interface the user is interacting with.

### 详细描述:

Web应用程序应该限制是否允许在框架，iframe，对象，嵌入或applet元素中呈现它。如果没有这些限制，用户可能会被欺骗与应用程序进行交互。

### Extended Description:

A web application is expected to place restrictions on whether it is allowed to be rendered within frames, iframes, objects, embed or applet elements. Without the restrictions, users can be tricked into interacting with the application when they were not intending to.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1022

提交日期 2018-03-27---> 修改日期 2018-03-27

* **Weakness Name:** *en: --> Use of Web Link to Untrusted Target with window.opener Access zh: -->使用window.opener Access将Web链接用于不受信任的目标*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

Web应用程序生成指向其控制范围之外的不受信任的外部站点的链接，但它不能正确地阻止外部站点修改window.opener对象的安全关键属性，例如location属性。

### Description:

The web application produces links to untrusted external sites outside of its sphere of control, but it does not properly prevent the external site from modifying security-critical properties of the window.opener object, such as the location property.

### 详细描述:

当用户单击指向外部站点（“目标”）的链接时，target =“\_ blank”属性会导致目标站点的内容在新窗口或选项卡中打开，该窗口或选项卡在与原始页面相同的进程中运行。 window.opener对象记录有关提供链接的原始页面的信息。如果攻击者可以在目标页面上运行脚本，那么他们可以读取或修改window.opener对象的某些属性，包括location属性 - 即使原始站点和目标站点不是同一个源。攻击者可以修改位置属性以自动将用户重定向到恶意站点，例如作为网络钓鱼攻击的一部分。由于此重定向发生在原始窗口/选项卡中 - 这不一定是可见的，因为浏览器将显示重点放在新目标页面上 - 用户可能不会注意到任何可疑的重定向。

### Extended Description:

When a user clicks a link to an external site ("target"), the target="\_blank" attribute causes the target site's contents to be opened in a new window or tab, which runs in the same process as the original page. The window.opener object records information about the original page that offered the link. If an attacker can run script on the target page, then they could read or modify certain properties of the window.opener object, including the location property - even if the original and target site are not the same origin. An attacker can modify the location property to automatically redirect the user to a malicious site, e.g. as part of a phishing attack. Since this redirect happens in the original window/tab - which is not necessarily visible, since the browser is focusing the display on the new target page - the user might not notice any suspicious redirection.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1023

提交日期 I`m don`t know---> 修改日期 2019-01-03

* **Weakness Name:** *en: --> Incomplete Comparison with Missing Factors zh: -->与缺失因素的不完全比较*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在必须考虑每个实体的多个因素或特征的实体之间进行比较，但该比较不包括这些因素中的一个或多个。这可能导致产生的弱点，例如，通过操作错误的对象。

### Description:

The software performs a comparison between entities that must consider multiple factors or characteristics of each entity, but the comparison does not include one or more of these factors. This can lead to resultant weaknesses, e.g. by operating on the wrong object.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1024

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Comparison of Incompatible Types zh: -->不相容类型的比较*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件执行两个实体之间的比较，但实体具有不同的，不兼容的类型，在直接比较时无法保证提供正确的结果。

### Description:

The software performs a comparison between two entities, but the entities are of different, incompatible types that cannot be guaranteed to provide correct results when they are directly compared.

### 详细描述:

在严格类型但支持转换/转换的语言中，例如C或C ++，程序员可能会假设将一个实体转换为与另一个实体相同的类型将确保比较将正确执行，但这不能保证。在没有严格类型的语言中，例如PHP或JavaScript，可能会隐式转换/转换为程序员不知道的类型，从而导致意外结果;例如，字符串“123”可能会转换为数字类型。见例子。

### Extended Description:

In languages that are strictly typed but support casting/conversion, such as C or C++, the programmer might assume that casting one entity to the same type as another entity will ensure that the comparison will be performed correctly, but this cannot be guaranteed. In languages that are not strictly typed, such as PHP or JavaScript, there may be implicit casting/conversion to a type that the programmer is unaware of, causing unexpected results; for example, the string "123" might be converted to a number type. See examples.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1025

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Comparison Using Wrong Factors zh: -->使用错误因素的比较*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件执行两个实体之间的比较，但比较检查实体的错误因素或特征，这可能导致不正确的结果和由此产生的弱点。

### Description:

The software performs a comparison between two entities, but the comparison examines the wrong factors or characteristics of the entities, which can lead to incorrect results and resultant weaknesses.

### 详细描述:

当代码无意中提取对对象的引用而不是其相关内容时，会出现这种弱点的常见示例。

### Extended Description:

A common example of this weakness occurs when the code inadvertently extracts the reference to an object, instead of its relevant contents.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1037

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Processor Optimization Removal or Modification of Security-critical Code zh: -->处理器优化删除或修改安全关键代码*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

开发人员在软件中构建安全关键保护机制，但处理器优化程序的执行，以便删除或修改机制。

### Description:

The developer builds a security-critical protection mechanism into the software, but the processor optimizes the execution of the program such that the mechanism is removed or modified.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1038

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Insecure Automated Optimizations zh: -->不安全的自动优化*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用一种自动优化代码的机制，例如改善性能等特性，但优化可能会产生可能违反预期安全假设的意外副作用。

### Description:

The product uses a mechanism that automatically optimizes code, e.g. to improve a characteristic such as performance, but the optimizations can have an unintended side effect that might violate an intended security assumption.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1039

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Automated Recognition Mechanism with Inadequate Detection or Handling of Adversarial Input Perturbations zh: -->具有不充分检测或处理对抗性输入扰动的自动识别机制*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用机器学习等自动机制将复杂的数据输入（例如图像或音频）识别为特定的概念或类别，但它没有正确检测或处理以导致机制的方式修改或构造的输入检测一个不同的，不正确的概念。

### Description:

The product uses an automated mechanism such as machine learning to recognize complex data inputs (e.g. image or audio) as a particular concept or category, but it does not properly detect or handle inputs that have been modified or constructed in a way that causes the mechanism to detect a different, incorrect concept.

### 详细描述:

当使用机器学习等技术自动对输入流进行分类，并将这些分类用于安全关键决策时，分类中的任何错误都可能引入一个漏洞，允许攻击者使产品做出错误的安全决策。如果自动机制没有开发或“训练”有足够的输入数据，那么攻击者可能能够制造故意触发错误分类的恶意输入。  
目标技术包括但不一定限于：  
  
自动语音识别  
自动图像识别  
  
例如，攻击者可能会修改道路标志或路面标记，以欺骗自动驾驶车辆误读标志/标记并执行危险行为。

### Extended Description:

When techniques such as machine learning are used to automatically classify input streams, and those classifications are used for security-critical decisions, then any mistake in classification can introduce a vulnerability that allows attackers to cause the product to make the wrong security decision. If the automated mechanism is not developed or "trained" with enough input data, then attackers may be able to craft malicious input that intentionally triggers the incorrect classification.  
Targeted technologies include, but are not necessarily limited to:  
  
automated speech recognition  
automated image recognition  
  
For example, an attacker might modify road signs or road surface markings to trick autonomous vehicles into misreading the sign/marking and performing a dangerous action.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

需要进一步调查以确定是否存在更好的关系或是否需要创建其他组织条目。例如，此问题可能更好地与“将输入识别为错误类型”相关，这可能会将其视为CWE-704（不正确的类型转换）的兄弟。

Weakness ID: 1041

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Use of Redundant Code zh: -->使用冗余代码*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件具有包含相同代码的多个功能，方法，过程，宏等。

### Description:

The software has multiple functions, methods, procedures, macros, etc. that contain the same code.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。例如，如果有两个相同代码的副本，程序员可能会修复一个副本中的弱点，而忘记在另一个副本中修复相同的弱点。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. For example, if there are two copies of the same code, the programmer might fix a weakness in one copy while forgetting to fix the same weakness in another copy.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1042

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Static Member Data Element outside of a Singleton Class Element zh: -->Singleton类元素之外的静态成员数据元素*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码包含一个声明为static（但不是final）的成员元素，其中父类元素不是单例类 - 也就是说，一个只能在Create的'to'关联中使用一次的类元素行动。

### Description:

The code contains a member element that is declared as static (but not final), in which its parent class element is not a singleton class - that is, a class element that can be used only once in the 'to' association of a Create action.

### 详细描述:

此问题可能会使软件执行速度变慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。

### Extended Description:

This issue can make the software perform more slowly. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1043

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Data Element Aggregating an Excessively Large Number of Non-Primitive Elements zh: -->数据元素聚合过多的非原始元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用具有过多数量的子元素的数据元素，其具有非原始数据类型，例如结构或聚合对象。

### Description:

The software uses a data element that has an excessively large number of sub-elements with non-primitive data types such as structures or aggregated objects.

### 详细描述:

此问题可能会使软件执行速度变慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。  
虽然“过大”的解释可能因每个产品或开发人员而异，但CISQ建议默认使用5个子元素。

### Extended Description:

This issue can make the software perform more slowly. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.  
While the interpretation of "excessively large" may vary for each product or developer, CISQ recommends a default of 5 sub-elements.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1044

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Architecture with Number of Horizontal Layers Outside of Expected Range zh: -->具有预期范围之外的水平层数的体系结构*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件的架构包含太多 - 或太少 - 水平层。

### Description:

The software's architecture contains too many - or too few - horizontal layers.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。  
虽然每个产品或开发人员对“预期范围”的解释可能有所不同，但CISQ建议默认最少4层，最多8层。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.  
While the interpretation of "expected range" may vary for each product or developer, CISQ recommends a default minimum of 4 layers and maximum of 8 layers.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1045

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Parent Class with a Virtual Destructor and a Child Class without a Virtual Destructor zh: -->具有虚拟析构函数和没有虚拟析构函数的子类的父类*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

父类具有虚拟析构函数方法，但父类具有没有虚析构函数的子类。

### Description:

A parent class has a virtual destructor method, but the parent has a child class that does not have a virtual destructor.

### 详细描述:

此问题可能会阻止软件可靠地运行，因为孩子可能无法执行必要的销毁操作。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞，例如内存泄漏（CWE-401）。

### Extended Description:

This issue can prevent the software from running reliably, since the child might not perform essential destruction operations. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability, such as a memory leak (CWE-401).

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1046

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Creation of Immutable Text Using String Concatenation zh: -->使用字符串连接创建不可变文本*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用字符串连接操作创建不可变的文本字符串。

### Description:

The software creates an immutable text string using string concatenation operations.

### 详细描述:

与使用文本缓冲数据元素相比，该编程模式可能是低效的。  
此问题可能会使软件执行速度变慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。

### Extended Description:

This programming pattern can be inefficient in comparison with use of text buffer data elements.  
This issue can make the software perform more slowly. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1047

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Modules with Circular Dependencies zh: -->具有循环依赖关系的模块*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含模块，其中一个模块具有循环回自身的引用，即存在循环依赖性。

### Description:

The software contains modules in which one module has references that cycle back to itself, i.e., there are circular dependencies.

### 详细描述:

作为一个例子，使用Java，这个弱点可能表明包之间的循环。  
由于模块化不足，这个问题使得维护软件变得更加困难，这通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。  
此问题可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

As an example, with Java, this weakness might indicate cycles between packages.  
This issue makes it more difficult to maintain the software due to insufficient modularity, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.  
This issue can prevent the software from running reliably. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1048

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Invokable Control Element with Large Number of Outward Calls zh: -->具有大量外拨呼叫的可调用控制元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码包含可调用的控制元素，这些控制元素包含对可调用的上下文外部的其他应用程序对象的过多数量的引用，即过大的扇出值。

### Description:

The code contains callable control elements that contain an excessively large number of references to other application objects external to the context of the callable, i.e. a Fan-Out value that is excessively large.

### 详细描述:

虽然每个产品或开发人员对“过大的扇出值”的解释可能有所不同，但CISQ建议默认使用5个引用对象。  
此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

While the interpretation of "excessively large Fan-Out value" may vary for each product or developer, CISQ recommends a default of 5 referenced objects.  
This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1049

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Data Query Operations in a Large Data Table zh: -->大数据表中过多的数据查询操作*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在大型数据表上执行大量连接和子查询的数据查询。

### Description:

The software performs a data query with a large number of joins and sub-queries on a large data table.

### 详细描述:

此问题可能会使软件执行速度变慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。  
虽然每个产品或开发人员对“大数据表”和“大量连接或子查询”的解释可能有所不同，但CISQ建议对“大”数据表默认为100万行，默认最小为5个连接，以及默认的最少3个子查询。

### Extended Description:

This issue can make the software perform more slowly. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.  
While the interpretation of "large data table" and "large number of joins or sub-queries" may vary for each product or developer, CISQ recommends a default of 1 million rows for a "large" data table, a default minimum of 5 joins, and a default minimum of 3 sub-queries.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1050

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Platform Resource Consumption within a Loop zh: -->循环中过多的平台资源消耗*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件具有循环体或循环条件，其包含直接或间接消耗平台资源的控制元素，例如，消息传递，会话，锁或文件描述符。

### Description:

The software has a loop body or loop condition that contains a control element that directly or indirectly consumes platform resources, e.g. messaging, sessions, locks, or file descriptors.

### 详细描述:

此问题可能会使软件执行速度变慢。如果攻击者可以影响循环中的迭代次数，那么此性能问题可能会通过消耗比预期更多的平台资源来允许拒绝服务。

### Extended Description:

This issue can make the software perform more slowly. If an attacker can influence the number of iterations in the loop, then this performance problem might allow a denial of service by consuming more platform resources than intended.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1051

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Initialization with Hard-Coded Network Resource Configuration Data zh: -->使用硬编码网络资源配置数据进行初始化*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用充当网络资源标识符的硬编码值初始化数据。

### Description:

The software initializes data using hard-coded values that act as as network resource identifiers.

### 详细描述:

此问题可能会阻止软件可靠地运行，例如如果它在环境中运行，则不使用硬编码的网络资源标识符。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

This issue can prevent the software from running reliably, e.g. if it runs in an environment does not use the hard-coded network resource identifiers. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1052

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Use of Hard-Coded Literals in Initialization zh: -->在初始化中过度使用硬编码文字*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用硬编码文字初始化数据元素，该文字不是简单的整数或静态常量元素。

### Description:

The software initializes a data element using a hard-coded literal that is not a simple integer or static constant element.

### 详细描述:

此问题使得修改或维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to modify or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1053

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Missing Documentation for Design zh: -->缺少设计文档*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品没有代表其设计方式的文档。

### Description:

The product does not have documentation that represents how it is designed.

### 详细描述:

此问题可能使理解和维护代码变得更加困难。它可以使检测和/或修复漏洞变得更加困难和耗时。

### Extended Description:

This issue can make it more difficult to understand and maintain the code. It can make it more difficult and time-consuming to detect and/or fix vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1054

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Invocation of a Control Element at an Unnecessarily Deep Horizontal Layer zh: -->在不必要的深水平层调用控制元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

一个架构层的代码调用驻留在比相邻层更深的层的代码，即，调用跳过至少一个层，并且调用的代码不是可以从任何水平层引用的垂直实用层的一部分。

### Description:

The code at one architectural layer invokes code that resides at a deeper layer than the adjacent layer, i.e., the invocation skips at least one layer, and the invoked code is not part of a vertical utility layer that can be referenced from any horizontal layer.

### 详细描述:

此问题使得理解和维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to understand and maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1055

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Multiple Inheritance from Concrete Classes zh: -->具体类的多重继承*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个具有多个具体类继承的类。

### Description:

The software contains a class with inheritance from more than one concrete class.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1056

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Invokable Control Element with Variadic Parameters zh: -->具有可变参数的可调控制元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

命名可调用或方法控制元素具有支持变量（可变参数）数量的参数或参数的签名。

### Description:

A named-callable or method control element has a signature that supports a variable (variadic) number of parameters or arguments.

### 详细描述:

此问题可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。  
对于可变参数，手动分析确定调用哪个函数/方法可能是困难的或低效的。

### Extended Description:

This issue can prevent the software from running reliably. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.  
With variadic arguments, it can be difficult or inefficient for manual analysis to be certain of which function/method is being invoked.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1057

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Data Access Operations Outside of Expected Data Manager Component zh: -->预期数据管理器组件之外的数据访问操作*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件根据设计要求使用专用的中央数据管理器组件，但它包含执行不使用此数据管理器的数据访问操作的代码。

### Description:

The software uses a dedicated, central data manager component as required by design, but it contains code that performs data-access operations that do not use this data manager.

### 详细描述:

此问题可能使软件执行速度比预期慢，因为预期的中央数据管理器可能已针对性能或其他质量特性进行了明确优化。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。

### Extended Description:

This issue can make the software perform more slowly than intended, since the intended central data manager may have been explicitly optimized for performance or other quality characteristics. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1058

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Invokable Control Element in Multi-Thread Context with non-Final Static Storable or Member Element zh: -->具有非最终静态可存储或成员元素的多线程上下文中的可调用控制元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码包含在多线程环境中运行但具有不安全的非最终静态可存储或成员数据元素的函数或方法。

### Description:

The code contains a function or method that operates in a multi-threaded environment but owns an unsafe non-final static storable or member data element.

### 详细描述:

此问题可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

This issue can prevent the software from running reliably. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1059

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Incomplete Documentation zh: -->不完整的文档*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

无论是纸质还是电子形式的文档都不包含产品所有相关元素的描述，例如其用法，结构，接口，设计，实现，配置，操作等。

### Description:

The documentation, whether on paper or in electronic form, does not contain descriptions of all the relevant elements of the product, such as its usage, structure, interfaces, design, implementation, configuration, operation, etc.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1060

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Number of Inefficient Server-Side Data Accesses zh: -->过多的低效服务器端数据访问次数*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件执行太多数据查询而不使用诸如存储过程之类的有效数据处理功能。

### Description:

The software performs too many data queries without using efficient data processing functionality such as stored procedures.

### 详细描述:

由于计算开销，此问题可能使软件执行速度更慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。  
虽然“太多数据查询”的交互可能因每个产品或开发人员而异，但CISQ建议默认最多5个数据查询用于低效的功能/过程。

### Extended Description:

This issue can make the software perform more slowly due to computational expense. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.  
While the interpetation of "too many data queries" may vary for each product or developer, CISQ recommends a default maximum of 5 data queries for an inefficient function/procedure.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1061

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Insufficient Encapsulation zh: -->封装不足*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件没有充分隐藏数据或方法的内部表示和实现细节，这可能允许外部组件或模块意外修改数据，调用意外功能或引入程序员不想要的依赖项。

### Description:

The software does not sufficiently hide the internal representation and implementation details of data or methods, which might allow external components or modules to modify data unexpectedly, invoke unexpected functionality, or introduce dependencies that the programmer did not intend.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1062

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Parent Class with References to Child Class zh: -->引用子类的父类*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码有一个父类，它包含对子类，方法或其成员的引用。

### Description:

The code has a parent class that contains references to a child class, its methods, or its members.

### 详细描述:

此问题可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

This issue can prevent the software from running reliably. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1063

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Creation of Class Instance within a Static Code Block zh: -->在静态代码块中创建类实例*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

静态代码块创建类的实例。

### Description:

A static code block creates an instance of a class.

### 详细描述:

此模式标识了使用声明为静态的代码块中的值初始化可存储数据元素或成员数据元素的情况。  
此问题可以通过在需要之前执行初始化来使软件执行得更慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。

### Extended Description:

This pattern identifies situations where a storable data element or member data element is initialized with a value in a block of code which is declared as static.  
This issue can make the software perform more slowly by performing initialization before it is needed. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1064

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Invokable Control Element with Signature Containing an Excessive Number of Parameters zh: -->具有包含过多参数的签名的可调用控制元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个函数，子例程或方法，其签名具有不必要的大量参数/参数。

### Description:

The software contains a function, subroutine, or method whose signature has an unnecessarily large number of parameters/arguments.

### 详细描述:

此问题使得理解和/或维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。  
同时插入“大量参数”。可能因每个产品或开发人员而异，CISQ建议默认最多7个参数/参数。

### Extended Description:

This issue makes it more difficult to understand and/or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.  
While the interpetation of "large number of parameters." may vary for each product or developer, CISQ recommends a default maximum of 7 parameters/arguments.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1065

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Runtime Resource Management Control Element in a Component Built to Run on Application Servers zh: -->构建为在应用程序服务器上运行的组件中的运行时资源管理控制元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序使用来自应用程序服务器的已部署组件，但它也使用低级功能/方法来管理资源，而不是应用程序服务器提供的API。

### Description:

The application uses deployed components from application servers, but it also uses low-level functions/methods for management of resources, instead of the API provided by the application server.

### 详细描述:

此问题可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

This issue can prevent the software from running reliably. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1066

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Missing Serialization Control Element zh: -->缺少序列化控制元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个可序列化的数据元素，该元素没有相关的序列化方法。

### Description:

The software contains a serializable data element that does not have an associated serialization method.

### 详细描述:

此问题可能会阻止软件可靠地运行，例如通过触发一个例子。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。  
例如，数据元素的可序列化特性来自.NET中的可序列化SerializableAttribute属性以及Java中java.io.Serializable接口的继承。

### Extended Description:

This issue can prevent the software from running reliably, e.g. by triggering an exeption. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.  
As examples, the serializable nature of a data element comes from a serializable SerializableAttribute attribute in .NET and the inheritance from the java.io.Serializable interface in Java.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1067

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Execution of Sequential Searches of Data Resource zh: -->过度执行数据资源的序列搜索*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含针对SQL表或视图的数据查询，该查询以不使用索引的方式配置，并可能导致执行顺序搜索。

### Description:

The software contains a data query against an SQL table or view that is configured in a way that does not utilize an index and may cause sequential searches to be performed.

### 详细描述:

此问题可能会使软件执行速度变慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。

### Extended Description:

This issue can make the software perform more slowly. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1068

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Inconsistency Between Implementation and Documented Design zh: -->实施与文件设计之间的不一致*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品的实施与相关文档中描述的设计不一致。

### Description:

The implementation of the product is not consistent with the design as described within the relevant documentation.

### 详细描述:

此问题使得由于不一致而更难以维护软件，这通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software due to inconsistencies, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1069

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Empty Exception Block zh: -->空例外块*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

可调用代码块包含不包含任何代码的异常处理块，即为空。

### Description:

An invokable code block contains an exception handling block that does not contain any code, i.e. is empty.

### 详细描述:

当使用异常处理块（例如Catch和Finally块）但该块为空时，这可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

When an exception handling block (such as a Catch and Finally block) is used, but that block is empty, this can prevent the software from running reliably. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1070

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Serializable Data Element Containing non-Serializable Item Elements zh: -->包含非可序列化项元素的可序列化数据元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含可序列化的可存储数据元素，如字段或成员，但数据元素包含不可序列化的成员元素。

### Description:

The software contains a serializable, storable data element such as a field or member, but the data element contains member elements that are not serializable.

### 详细描述:

此问题可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。  
例如，数据元素的可序列化特性来自.NET中的可序列化SerializableAttribute属性以及Java中java.io.Serializable接口的继承。

### Extended Description:

This issue can prevent the software from running reliably. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.  
As examples, the serializable nature of a data element comes from a serializable SerializableAttribute attribute in .NET and the inheritance from the java.io.Serializable interface in Java.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1071

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Empty Code Block zh: -->空代码块*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

源代码包含一个不包含任何代码的块，即块是空的。

### Description:

The source code contains a block that does not contain any code, i.e., the block is empty.

### 详细描述:

空代码块可以出现在条件，函数或方法定义，异常处理程序等的主体中。虽然空代码块可能是有意的，但它也可能表示不完整的实现，意外的代码删除，意外的宏扩展等。对于某些编程语言和构造，语法可能允许空块，但块中缺少任何行为可能违反约定或API，使其成为错误。

### Extended Description:

Empty code blocks can occur in the bodies of conditionals, function or method definitions, exception handlers, etc. While an empty code block might be intentional, it might also indicate incomplete implementation, accidental code deletion, unexpected macro expansion, etc. For some programming languages and constructs, an empty block might be allowed by the syntax, but the lack of any behavior within the block might violate a convention or API in such a way that it is an error.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1072

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Data Resource Access without Use of Connection Pooling zh: -->不使用连接池的数据资源访问*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件通过数据库访问数据资源，而不使用连接池功能。

### Description:

The software accesses a data resource through a database without using a connection pooling capability.

### 详细描述:

此问题可能会使软件执行速度变慢，因为连接池允许重用连接，而无需打开和关闭新连接的开销和时间消耗。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。

### Extended Description:

This issue can make the software perform more slowly, as connection pools allow connections to be reused without the overhead and time consumption of opening and closing a new connection. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1073

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Non-SQL Invokable Control Element with Excessive Number of Data Resource Accesses zh: -->具有过多数据资源访问的非SQL可调度控制元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含具有功能或方法的客户机，该功能或方法包含通过数据管理器发送的大量数据访问/查询，即，不使用有效的数据库功能。

### Description:

The software contains a client with a function or method that contains a large number of data accesses/queries that are sent through a data manager, i.e., does not use efficient database capabilities.

### 详细描述:

此问题可能会使软件执行速度变慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。  
虽然“大量数据访问/查询”的交互可能因每个产品或开发人员而异，但CISQ建议每个功能/方法默认最多2个数据访问。

### Extended Description:

This issue can make the software perform more slowly. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.  
While the interpetation of "large number of data accesses/queries" may vary for each product or developer, CISQ recommends a default maximum of 2 data accesses per function/method.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1074

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Class with Excessively Deep Inheritance zh: -->具有过度深度继承的类*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

类具有过高的继承级别，即它具有大量父类。

### Description:

A class has an inheritance level that is too high, i.e., it has a large number of parent classes.

### 详细描述:

此问题使得理解和维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。  
虽然“大量父类”的交互可能因每个产品或开发人员而异，但CISQ建议默认最多7个父类。

### Extended Description:

This issue makes it more difficult to understand and maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.  
While the interpetation of "large number of parent classes" may vary for each product or developer, CISQ recommends a default maximum of 7 parent classes.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1075

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Unconditional Control Flow Transfer outside of Switch Block zh: -->交换机块外的无条件控制流转移*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在诸如开关块之类的分支结构之外的代码中执行无条件控制传输（例如“goto”）。

### Description:

The software performs unconditional control transfer (such as a "goto") in code outside of a branching structure such as a switch block.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1076

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Insufficient Adherence to Expected Conventions zh: -->对遵守预期公约的遵守不足*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品的体系结构，源代码，设计，文档或其他工件不遵循必需的约定。

### Description:

The product's architecture, source code, design, documentation, or other artifact does not follow required conventions.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1077

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Floating Point Comparison with Incorrect Operator zh: -->与不正确的算子的浮点比较*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码执行比较，例如两个浮点（浮点）值之间的相等性测试，但它使用不考虑精度损失可能性的比较运算符。

### Description:

The code performs a comparison such as an equality test between two float (floating point) values, but it uses comparison operators that do not account for the possibility of loss of precision.

### 详细描述:

使用浮点值进行数值计算  
由于舍入错误，可能会生成不精确的结果。  
结果，可能产生两种不同的计算  
在数学上相等，但略有不同的数字  
不转换为的不同位表示  
相同的数学上相等的值。结果，平等  
测试或其他比较可能会产生意外  
结果。  
此问题可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

Numeric calculation using floating point values  
 can generate imprecise results because of rounding errors.  
 As a result, two different calculations might generate  
 numbers that are mathematically equal, but have slightly  
 different bit representations that do not translate to the  
 same mathematically-equal values. As a result, an equality  
 test or other comparison might produce unexpected  
 results.  
This issue can prevent the software from running reliably. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1078

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Inappropriate Source Code Style or Formatting zh: -->不恰当的源代码样式或格式*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

源代码不符合缩进，空格，注释等所需的样式或格式。

### Description:

The source code does not follow desired style or formatting for indentation, white space, comments, etc.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1079

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Parent Class without Virtual Destructor Method zh: -->没有虚拟析构函数方法的父类*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

父类包含一个或多个子类，但父类没有虚拟析构函数方法。

### Description:

A parent class contains one or more child classes, but the parent class does not have a virtual destructor method.

### 详细描述:

此问题可能会阻止软件由于未定义或意外行为而可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

This issue can prevent the software from running reliably due to undefined or unexpected behaviors. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1080

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Source Code File with Excessive Number of Lines of Code zh: -->具有过多代码行的源代码文件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

源代码文件包含太多代码行。

### Description:

A source code file has too many lines of code.

### 详细描述:

此问题使得理解和/或维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。  
对于每个产品或开发人员而言，“太多代码行”的交叉可能会有所不同，CISQ建议默认阈值为1000。

### Extended Description:

This issue makes it more difficult to understand and/or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.  
Whilte the interpetation of "too many lines of code" may vary for each product or developer, CISQ recommends a default threshold value of 1000.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1082

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Class Instance Self Destruction Control Element zh: -->类实例自毁控制元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码包含一个类实例，它调用方法或函数来删除或销毁自身。

### Description:

The code contains a class instance that calls the method or function to delete or destroy itself.

### 详细描述:

例如，在C ++中，“删除此”将导致对象自行删除。  
此问题可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

For example, in C++, "delete this" will cause the object to delete itself.  
This issue can prevent the software from running reliably. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1083

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Data Access from Outside Expected Data Manager Component zh: -->来自外部预期数据管理器组件的数据访问*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件旨在通过特定数据管理器组件（如关系数据库或非SQL数据库）管理数据访问，但它包含执行数据访问操作而不使用该组件的代码。

### Description:

The software is intended to manage data access through a particular data manager component such as a relational or non-SQL database, but it contains code that performs data access operations without using that component.

### 详细描述:

当软件具有数据访问组件时，该设计可旨在处理通过该组件的所有数据访问操作。如果在该组件之外执行数据访问操作，则这可能表示违反了预期的设计。  
此问题可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

When the software has a data access component, the design may be intended to handle all data access operations through that component. If a data access operation is performed outside of that component, then this may indicate a violation of the intended design.  
This issue can prevent the software from running reliably. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1084

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Invokable Control Element with Excessive File or Data Access Operations zh: -->具有过多文件或数据访问操作的可调用控制元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

函数或方法包含太多使用数据管理器或文件资源的操作。

### Description:

A function or method contains too many operations that utilize a data manager or file resource.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。  
虽然每个产品或开发人员对“太多操作”的交互可能会有所不同，但CISQ建议对同一数据管理器或文件默认最多7个操作。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.  
While the interpetation of "too many operations" may vary for each product or developer, CISQ recommends a default maximum of 7 operations for the same data manager or file.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1085

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Invokable Control Element with Excessive Volume of Commented-out Code zh: -->具有过量注释代码的可调控制元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

函数，方法，过程等包含在其体内注释掉的过多代码。

### Description:

A function, method, procedure, etc. contains an excessive amount of code that has been commented out within its body.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。  
虽然每个产品或开发人员的“过量”的交集可能会有所不同，但CISQ建议默认阈值为注释代码的2％。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.  
While the interpetation of "excessive volume" may vary for each product or developer, CISQ recommends a default threshold of 2% of commented code.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1086

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Class with Excessive Number of Child Classes zh: -->具有过多子类的类*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

一个类包含不必要的大量子节点。

### Description:

A class contains an unnecessarily large number of children.

### 详细描述:

此问题使得理解和维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。  
虽然每个产品或开发人员对“大量儿童”的交互可能有所不同，但CISQ建议默认最多10个子类。

### Extended Description:

This issue makes it more difficult to understand and maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.  
While the interpetation of "large number of children" may vary for each product or developer, CISQ recommends a default maximum of 10 child classes.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1087

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Class with Virtual Method without a Virtual Destructor zh: -->具有虚拟方法的类没有虚拟析构函数*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

类包含虚方法，但该方法没有关联的虚析构函数。

### Description:

A class contains a virtual method, but the method does not have an associated virtual destructor.

### 详细描述:

此问题可能会阻止软件可靠地运行，例如由于未定义的行为。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

This issue can prevent the software from running reliably, e.g. due to undefined behavior. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1088

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Synchronous Access of Remote Resource without Timeout zh: -->远程资源的同步访问没有超时*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码具有对远程资源的同步调用，但调用没有超时，或者超时设置为无限。

### Description:

The code has a synchronous call to a remote resource, but there is no timeout for the call, or the timeout is set to infinite.

### 详细描述:

此问题可能会阻止软件可靠地运行，因为远程资源的中断可能导致软件挂起。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

This issue can prevent the software from running reliably, since an outage for the remote resource can cause the software to hang. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1089

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Large Data Table with Excessive Number of Indices zh: -->具有过多指数的大数据表*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用包含过多索引的大型数据表。

### Description:

The software uses a large data table that contains an excessively large number of indices.

### 详细描述:

此问题可能会使软件执行速度变慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。  
虽然“大数据表”和“过多索引”的交互可能因每个产品或开发人员而异，但CISQ建议“大”表的默认阈值为1000000行，默认阈值为3个索引。

### Extended Description:

This issue can make the software perform more slowly. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.  
While the interpetation of "large data table" and "excessively large number of indices" may vary for each product or developer, CISQ recommends a default threshold of 1000000 rows for a "large" table and a default threshold of 3 indices.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1090

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Method Containing Access of a Member Element from Another Class zh: -->包含从另一个类访问成员元素的方法*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

类的方法执行直接访问来自另一个类的成员元素的操作。

### Description:

A method for a class performs an operation that directly accesses a member element from another class.

### 详细描述:

这个问题表明封装不良并且使得理解和维护软件变得更加困难，这通过使查找和/或修复漏洞更加困难或耗时而间接地影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue suggests poor encapsulation and makes it more difficult to understand and maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1091

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Use of Object without Invoking Destructor Method zh: -->在不调用析构函数方法的情况下使用Object*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含一个访问对象的方法，但以后不会调用该元素的关联的finalize /析构函数方法。

### Description:

The software contains a method that accesses an object but does not later invoke the element's associated finalize/destructor method.

### 详细描述:

通过保留超过必要的内存和/或其他资源，此问题可以使软件执行速度更慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。

### Extended Description:

This issue can make the software perform more slowly by retaining memory and/or other resources longer than necessary. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1092

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Use of Same Invokable Control Element in Multiple Architectural Layers zh: -->在多个建筑图层中使用相同的Invokable控件元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件在多个体系结构层中使用相同的控制元素。

### Description:

The software uses the same control element across multiple architectural layers.

### 详细描述:

此问题使得理解和维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to understand and maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1093

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessively Complex Data Representation zh: -->过于复杂的数据表示*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件对其数据结构或这些结构之间的相互关系使用不必要的复杂内部表示。

### Description:

The software uses an unnecessarily complex internal representation for its data structures or interrelationships between those structures.

### 详细描述:

此问题使得理解或维护软件变得更加困难，通过使查找和/或修复漏洞更加困难或耗时，间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to understand or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1094

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Index Range Scan for a Data Resource zh: -->数据资源的索引范围扫描过多*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件包含大型数据表的索引范围扫描，但扫描可以覆盖大量行。

### Description:

The software contains an index range scan for a large data table, but the scan can cover a large number of rows.

### 详细描述:

此问题可能会使软件执行速度变慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。  
虽然“大数据表”和“过多索引范围”的交互可能因每个产品或开发人员而异，但CISQ建议阈值为1000000表行，并且索引范围的阈值为10。

### Extended Description:

This issue can make the software perform more slowly. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.  
While the interpetation of "large data table" and "excessive index range" may vary for each product or developer, CISQ recommends a threshold of 1000000 table rows and a threshold of 10 for the index range.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1095

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Loop Condition Value Update within the Loop zh: -->循环中的循环条件值更新*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用具有控制流条件的循环，该循环条件基于在循环体内更新的值。

### Description:

The software uses a loop with a control flow condition based on a value that is updated within the body of the loop.

### 详细描述:

此问题使得理解和/或维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to understand and/or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1096

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Singleton Class Instance Creation without Proper Locking or Synchronization zh: -->没有正确锁定或同步的单例类实例创建*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件实现了Singleton设计模式，但没有使用适当的锁定或其他同步机制来确保单例类仅实例化一次。

### Description:

The software implements a Singleton design pattern but does not use appropriate locking or other synchronization mechanism to ensure that the singleton class is only instantiated once.

### 详细描述:

此问题可能会阻止软件可靠地运行，例如通过使instantion进程非线程安全并引入死锁（CWE-833）或活锁条件。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

This issue can prevent the software from running reliably, e.g. by making the instantion process non-thread-safe and introducing deadlock (CWE-833) or livelock conditions. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1097

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Persistent Storable Data Element without Associated Comparison Control Element zh: -->具有相关比较控制元素的持久可存储数据元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用可存储的数据元素，该数据元素不具有支持比较所需的所有相关功能或方法。

### Description:

The software uses a storable data element that does not have all of the associated functions or methods that are necessary to support comparison.

### 详细描述:

例如，对于Java，持久化的类需要定义hashCode（）和equals（）方法。  
由于不正确或意外的比较结果，此问题可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

For example, with Java, a class that is made persistent requires both hashCode() and equals() methods to be defined.  
This issue can prevent the software from running reliably, due to incorrect or unexpected comparison results. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1098

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Data Element containing Pointer Item without Proper Copy Control Element zh: -->包含没有正确复制控制元素的指针项的数据元素*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码包含一个数据元素，其指针没有关联的副本或构造函数方法。

### Description:

The code contains a data element with a pointer that does not have an associated copy or constructor method.

### 详细描述:

此问题可能会阻止软件可靠地运行。如果攻击者可以访问相关代码，则此可靠性问题可能会引入漏洞。

### Extended Description:

This issue can prevent the software from running reliably. If the relevant code is reachable by an attacker, then this reliability problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1099

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Inconsistent Naming Conventions for Identifiers zh: -->标识符的命名约定不一致*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品的代码，文档或其他工件不会对变量，可调用项，相关可调用项组，I / O功能，数据类型，文件名或类似类型的元素使用相同的命名约定。

### Description:

The product's code, documentation, or other artifacts do not consistently use the same naming conventions for variables, callables, groups of related callables, I/O capabilities, data types, file names, or similar types of elements.

### 详细描述:

该问题使得由于不一致而更难以理解和/或维护软件，这通过使查找和/或修复漏洞更加困难或耗时而间接地影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to understand and/or maintain the software due to inconsistencies, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1100

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Insufficient Isolation of System-Dependent Functions zh: -->系统相关函数的隔离不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品或代码不会将与系统相关的功能隔离到单独的独立模块中。

### Description:

The product or code does not isolate system-dependent functionality into separate standalone modules.

### 详细描述:

此问题使得维护和/或移植软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain and/or port the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1101

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Reliance on Runtime Component in Generated Code zh: -->依赖于生成代码中的运行时组件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用自动生成的代码，如果没有特定的运行时支持组件，则无法执行。

### Description:

The product uses automatically-generated code that cannot be executed without a specific runtime support component.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1102

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Reliance on Machine-Dependent Data Representation zh: -->依赖于机器相关的数据表示*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码使用依赖于低级数据表示的数据表示或可能在不同处理器，物理机，操作系统或其他物理组件之间变化的构造。

### Description:

The code uses a data representation that relies on low-level data representation or constructs that may vary across different processors, physical machines, OSes, or other physical components.

### 详细描述:

此问题使得维护和/或移植软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain and/or port the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1103

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Use of Platform-Dependent Third Party Components zh: -->使用与平台相关的第三方组件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品依赖于第三方软件组件，这些组件不能在所有需要的平台上提供相同的功能。

### Description:

The product relies on third-party software components that do not provide equivalent functionality across all desirable platforms.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1104

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Use of Unmaintained Third Party Components zh: -->使用未维护的第三方组件*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品依赖于原始开发人员不支持或维护的第三方组件或原始开发人员的可信代理。

### Description:

The product relies on third-party components that are not actively supported or maintained by the original developer or a trusted proxy for the original developer.

### 详细描述:

依赖不再维护的组件可能很难或无法修复重大错误，漏洞或质量问题。实际上，未维护的代码可能会过时。  
此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

Reliance on components that are no longer maintained can make it difficult or impossible to fix significant bugs, vulnerabilities, or quality issues. In effect, unmaintained code can become obsolete.  
This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1105

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Insufficient Encapsulation of Machine-Dependent Functionality zh: -->机器相关功能的封装不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品或代码使用与机器相关的功能，但它没有充分封装或隔离此功能与其余代码。

### Description:

The product or code uses machine-dependent functionality, but it does not sufficiently encapsulate or isolate this functionality from the rest of the code.

### 详细描述:

此问题使得移植或维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to port or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1106

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Insufficient Use of Symbolic Constants zh: -->符号常量的使用不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

源代码使用可能需要随时间变化或演变的文字常量，而不是使用符号常量。

### Description:

The source code uses literal constants that may need to change or evolve over time, instead of using symbolic constants.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1107

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Insufficient Isolation of Symbolic Constant Definitions zh: -->符号常数定义的隔离不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

源代码使用符号常量，但它没有充分将这些常量的定义放入更集中或隔离的位置。

### Description:

The source code uses symbolic constants, but it does not sufficiently place the definitions of these constants into a more centralized or isolated location.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1108

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Reliance on Global Variables zh: -->过度依赖全球变量*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码的结构过于依赖于在代码中的各个点上使用或设置全局变量，而不是在更窄，更本地的上下文中保留相关信息。

### Description:

The code is structured in a way that relies too much on using or setting global variables throughout various points in the code, instead of preserving the associated information in a narrower, more local context.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1109

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Use of Same Variable for Multiple Purposes zh: -->为多重目的使用相同的变量*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码包含可调用的，块或其他代码元素，其中相同的变量用于控制多个唯一任务或存储多个数据实例。

### Description:

The code contains a callable, block, or other code element in which the same variable is used to control more than one unique task or store more than one instance of data.

### 详细描述:

将相同变量用于多种目的可能使人更难以阅读或理解代码，从而可能隐藏其他质量问题。  
此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

Use of the same variable for multiple purposes can make it more difficult for a person to read or understand the code, potentially hiding other quality issues.  
This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1110

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Incomplete Design Documentation zh: -->不完整的设计文档*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品的设计文档没有充分描述控制流程，数据流，系统初始化，任务之间的关系，组件，基本原理或设计的其他重要方面。

### Description:

The product's design documentation does not adequately describe control flow, data flow, system initialization, relationships between tasks, components, rationales, or other important aspects of the design.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1111

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Incomplete I/O Documentation zh: -->不完整的I / O文档*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

产品的文档没有充分定义输入，输出或系统/软件接口。

### Description:

The product's documentation does not adequately define inputs, outputs, or system/software interfaces.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1112

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Incomplete Documentation of Program Execution zh: -->程序执行的不完整文档*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该文档未完全定义用于控制或影响产品特定程序执行方式的所有机制。

### Description:

The document does not fully define all mechanisms that are used to control or influence how product-specific programs are executed.

### 详细描述:

这包括环境变量，配置文件，注册表项，命令行开关或选项或系统设置。

### Extended Description:

This includes environmental variables, configuration files, registry keys, command-line switches or options, or system settings.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1113

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Inappropriate Comment Style zh: -->不恰当的评论风格*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

源代码使用不一致或不符合产品预期标准的注释样式或格式。

### Description:

The source code uses comment styles or formats that are inconsistent or do not follow expected standards for the product.

### 详细描述:

由于可读性不足而导致维护软件更加困难，因此通过使查找和/或修复漏洞更加困难或耗时，间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software due to insufficient legibility, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1114

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Inappropriate Whitespace Style zh: -->不合适的空白风格*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

源代码包含在代码中不一致的空格，或者不符合产品的预期标准。

### Description:

The source code contains whitespace that is inconsistent across the code or does not follow expected standards for the product.

### 详细描述:

此问题使得理解和维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to understand and maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1115

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Source Code Element without Standard Prologue zh: -->没有标准序言的源代码元素*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

源代码包含诸如源文件之类的元素，这些元素不能始终如一地提供已为项目标准化的序言或标题。

### Description:

The source code contains elements such as source files that do not consistently provide a prologue or header that has been standardized for the project.

### 详细描述:

缺乏序言可能使准确和快速理解相关代码变得更加困难。标准序言或标题可能包含模块名称，版本号，作者，日期，目的，功能，假设，限制，准确性考虑等信息。  
由于可分析性不足，此问题使得维护软件变得更加困难，这通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

The lack of a prologue can make it more difficult to accurately and quickly understand the associated code. Standard prologues or headers may contain information such as module name, version number, author, date, purpose, function, assumptions, limitations, accuracy considerations, etc.  
This issue makes it more difficult to maintain the software due to insufficient analyzability, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1116

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Inaccurate Comments zh: -->不准确的评论*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

源代码包含的注释不能准确描述或解释与注释关联的代码部分的各个方面。

### Description:

The source code contains comments that do not accurately describe or explain aspects of the portion of the code with which the comment is associated.

### 详细描述:

当评论没有准确反映相关的代码元素时，这可能会给审阅者带来混淆（由于不一致），或者使得验证代码正确实现预期行为变得更加困难和效率更低。  
此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

When a comment does not accurately reflect the associated code elements, this can introduce confusion to a reviewer (due to inconsistencies) or make it more difficult and less efficient to validate that the code is implementing the intended behavior correctly.  
This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1117

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Callable with Insufficient Behavioral Summary zh: -->可用行为概述不足来调用*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码包含一个函数或方法，其签名和/或相关的内联文档无法充分描述可调用的输入，输出，副作用，假设或返回代码。

### Description:

The code contains a function or method whose signature and/or associated inline documentation does not sufficiently describe the callable's inputs, outputs, side effects, assumptions, or return codes.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1118

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Insufficient Documentation of Error Handling Techniques zh: -->错误处理技术的文档不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该文档没有充分描述用于错误处理，异常处理或类似机制的技术。

### Description:

The documentation does not sufficiently describe the techniques that are used for error handling, exception processing, or similar mechanisms.

### 详细描述:

文档可能需要涵盖多个层的错误处理技术，例如模块，可执行文件，可编译代码单元或可调用。

### Extended Description:

Documentation may need to cover error handling techniques at multiple layers, such as module, executable, compilable code unit, or callable.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1119

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Use of Unconditional Branching zh: -->过度使用无条件分支*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码使用了太多无条件分支（例如“goto”）。

### Description:

The code uses too many unconditional branches (such as "goto").

### 详细描述:

此问题使得理解和/或维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to understand and/or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1120

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Code Complexity zh: -->过多的代码复杂性*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码太复杂，使用定义明确的定量测量计算。

### Description:

The code is too complex, as calculated using a well-defined, quantitative measure.

### 详细描述:

此问题使得理解和/或维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。  
此问题可能会使软件执行速度变慢。如果攻击者可以访问相关代码，则此性能问题可能会引入漏洞。

### Extended Description:

This issue makes it more difficult to understand and/or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.  
This issue can make the software perform more slowly. If the relevant code is reachable by an attacker, then this performance problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1121

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive McCabe Cyclomatic Complexity zh: -->过度的McCabe Cyclomatic复杂性*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码包含超过理想最大值的McCabe圈复杂度。

### Description:

The code contains McCabe cyclomatic complexity that exceeds a desirable maximum.

### 详细描述:

此问题使得理解和/或维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to understand and/or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1122

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Halstead Complexity zh: -->过度的Halstead复杂性*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该代码的结构使得Halstead复杂性度量超过期望的最大值。

### Description:

The code is structured in a way that a Halstead complexity measure exceeds a desirable maximum.

### 详细描述:

存在各种Halstead复杂性度量，例如程序词汇量大小或量。  
此问题使得理解和/或维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

A variety of Halstead complexity measures exist, such as program vocabulary size or volume.  
This issue makes it more difficult to understand and/or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1123

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Use of Self-Modifying Code zh: -->过度使用自修改代码*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品使用过多的自修改代码。

### Description:

The product uses too much self-modifying code.

### 详细描述:

此问题使得理解或维护软件变得更加困难，通过使查找和/或修复漏洞更加困难或耗时，间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to understand or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1124

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessively Deep Nesting zh: -->过度深度嵌套*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

代码包含可调用或其他代码分组，其中嵌套/分支太深。

### Description:

The code contains a callable or other code grouping in which the nesting / branching is too deep.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1125

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Excessive Attack Surface zh: -->过度攻击面*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该产品具有攻击面，其定量测量超过所需的最大值。

### Description:

The product has an attack surface whose quantitative measurement exceeds a desirable maximum.

### 详细描述:

源于软件安全性，“攻击面”度量通常反映不可信方（即潜在攻击者）可以使用的输入点和输出点的数量。较大的攻击面提供了更多的攻击场所，并为开发人员提供了更多引入漏洞的机会。在某些情况下，这项措施可能反映出安全以外的其他质量方面;例如，具有许多输入和输出的产品可能需要大量测试以改善代码覆盖。

### Extended Description:

Originating from software security, an "attack surface" measure typically reflects the number of input points and output points that can be utilized by an untrusted party, i.e. a potential attacker. A larger attack surface provides more places to attack, and more opportunities for developers to introduce weaknesses. In some cases, this measure may reflect other aspects of quality besides security; e.g., a product with many inputs and outputs may require a large number of tests in order to improve code coverage.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1126

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Declaration of Variable with Unnecessarily Wide Scope zh: -->具有不必要的宽范围的变量声明*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

源代码在一个范围内声明一个变量，但该变量仅在较窄的范围内使用。

### Description:

The source code declares a variable in one scope, but the variable is only used within a narrower scope.

### 详细描述:

此问题使得理解和/或维护软件变得更加困难，这会通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to understand and/or maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1127

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Compilation with Insufficient Warnings or Errors zh: -->编译时警告或错误不足*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

编译代码时没有启用足够的警告，这可能会阻止检测到细微的错误或质量问题。

### Description:

The code is compiled without sufficient warnings enabled, which may prevent the detection of subtle bugs or quality issues.

### 详细描述:

此问题使得维护软件变得更加困难，因为软件通过使查找和/或修复漏洞更加困难或耗时而间接影响安全性。它也可能使引入漏洞更容易。

### Extended Description:

This issue makes it more difficult to maintain the software, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1164

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Irrelevant Code zh: -->不相关的代码*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序包含对执行不重要的代码，即不进行状态更改并且没有改变数据或控制流的副作用，因此删除代码对功能或正确性没有影响。

### Description:

The program contains code that is not essential for execution, i.e. makes no state changes and has no side effects that alter data or control flow, such that removal of the code would have no impact to functionality or correctness.

### 详细描述:

不相关的代码可能包括死代码，  
未使用的初始化，空块，可能完全的代码  
因优化等原因删除

### Extended Description:

Irrelevant code could include dead code,  
 initialization that is not used, empty blocks, code that could be entirely  
 removed due to optimization, etc.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1173

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Improper Use of Validation Framework zh: -->错误使用验证框架*
* **Abstraction:** *Base* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

应用程序不使用或错误地使用源语言或独立库提供的输入验证框架。

### Description:

The application does not use, or incorrectly uses, an input validation framework that is provided by the source language or an independent library.

### 详细描述:

许多现代编码语言为开发人员提供了输入验证框架，使输入验证任务更容易，更不容易出错。这些框架将根据指定的条件自动检查所有输入，并在收到无效输入时直接执行错误处理程序。这些框架的不正确使用（即，不正确的实现或完全丢失）不能直接利用，但如果在应用程序中稍后不执行适当的输入验证，则可能导致可利用的情况。不使用提供的输入验证框架也会损害代码的可维护性，因为未来的开发人员可能无法识别用于代替验证框架的下游输入验证。

### Extended Description:

Many modern coding languages provide developers with input validation frameworks to make the task of input validation easier and less error-prone. These frameworks will automatically check all input against specified criteria and direct execution to error handlers when invalid input is received. The improper use (i.e., an incorrect implementation or missing altogether) of these frameworks is not directly exploitable, but can lead to an exploitable condition if proper input validation is not performed later in the application. Not using provided input validation frameworks can also hurt the maintainability of code as future developers may not recognize the downstream input validation being used in the place of the validation framework.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1174

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> ASP.NET Misconfiguration: Improper Model Validation zh: -->ASP.NET配置错误：模型验证不正确*
* **Abstraction:** *Variant* **Structure:** *Simple* **Status:** *Draft*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

ASP.NET应用程序不使用或错误地使用模型验证框架。

### Description:

The ASP.NET application does not use, or incorrectly uses, the model validation framework.

### 详细描述:

没有详细描述

### Extended Description:

没有详细描述

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1176

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Inefficient CPU Computation zh: -->CPU计算效率低下*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该程序使用的算法执行CPU计算，这些算法不如开发人员的需要那样有效，即，可以进一步优化计算。

### Description:

The program performs CPU computations using algorithms that are not as efficient as they could be for the needs of the developer, i.e., the computations can be optimized further.

### 详细描述:

此问题可能使软件执行速度变慢，可能以用户可察觉的方式执行。如果攻击者可以影响必须执行的计算量，例如通过触发最坏情况的复杂性，这个性能问题可能会引入漏洞。

### Extended Description:

This issue can make the software perform more slowly, possibly in ways that are noticeable to the users. If an attacker can influence the amount of computation that must be performed, e.g. by triggering worst-case complexity, then this performance problem might introduce a vulnerability.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记

Weakness ID: 1177

提交日期 I`m don`t know---> 修改日期 I`m don`t know

* **Weakness Name:** *en: --> Use of Prohibited Code zh: -->禁止使用代码*
* **Abstraction:** *Class* **Structure:** *Simple* **Status:** *Incomplete*
* **Applicable\_Platforms:** *没有应用平台*

### 简单描述:

该软件使用已明确禁止的功能，库或第三方组件，无论是开发人员还是客户。

### Description:

The software uses a function, library, or third party component that has been explicitly prohibited, whether by the developer or the customer.

### 详细描述:

开发人员或客户可能出于各种原因（包括真实或可疑的漏洞）限制或消除功能，库或第三方组件的使用;难以安全使用;出口管制或许可证要求;代码过时或维护不善;内部代码被安排弃用;等等  
为了降低漏洞风险，开发人员可能会维护一个程序员必须避免使用的“禁止”功能列表，因为这些功能很难或不可能安全使用。此问题还可能使软件成本更高且难以维护。

### Extended Description:

The developer - or customers - may wish to restrict or eliminate use of a function, library, or third party component for any number of reasons, including real or suspected vulnerabilities; difficulty to use securely; export controls or license requirements; obsolete or poorly-maintained code; internal code being scheduled for deprecation; etc.  
To reduce risk of vulnerabilities, the developer might maintain a list of "banned" functions that programmers must avoid using because the functions are difficult or impossible to use securely. This issue can also make the software more costly and difficult to maintain.

### 问题背景 (Background Detail):

没有问题背景

### 笔记 (Notes):

没有笔记