代码安全编写规范

1. 安全编码

1.1. 通用编码原则

(一) 不要信任外部的用户输入或系统。

应用程序应该彻底验证所有用户输入，然后再根据用户输入执行操作。验证可能包括筛选特殊字符。针对用户意外地错误使用和某些人通过在系统中注入恶意命令蓄意进行攻击的情况，这种预防性措施对应用程序起到了保护作用。常见的例子包括 SQL 注入攻击、脚本注入和缓冲区溢出。此外，对于任何非受控的外部系统，都不要假定其安全性。

(二) 不要通过隐藏来保障安全。

尝试使用让人迷惑的变量名来隐藏机密信息或将它们存储在不常用的文件位置，这些方法都不能提供安全保障，最好使用平台功能或使用已被证实可行的技术来保护数据。

(三) 以安全的方式处理失效

如果应用程序失效（如发生严重错误等），要恰当的进行处理，一定要保护好机密数据。同时，在向最终用户返回错误消息时，不要公开任何不需要公开的信息。也就是不要提供任何有助于攻击者发现应用程序漏洞的详细信息。

1.2. 防范常见安全编码问题

在实现应用软件的编码阶段，也较容易因缺乏严谨思考或不好的编程习惯而引入安全问题，而且这些安全问题产生的危害作用非常大，因其产生的漏洞常常会造成应用程序中其他部分构筑的安全控制措施完全失效. 目前存在的相当数量系统漏洞都是由编码问题造成的. 因此要想保证应用软件的安全性，必须在编码阶段继续高度贯彻安全性原则.

在编码阶段，避免安全问题的基本原则如下：

- 程序只实现指定的功能

- 永远不要信任用户输入，对用户输入数据做有效性检查
- 必须考虑意外情况并进行处理
- 不要试图在发现错误之后继续执行
- 尽可能使用安全函数进行编程
- 小心、认真、细致地编程

目前在各种应用软件中常见的安全漏洞如下所示，应对这些常见问题进行有针对性的防范。

1.2.1 缓冲区溢出

如果对输入参数（字符串、整数等）处理时长度检查不严格，或对指针和数组越界访问不进行保护，就容易产生缓冲区溢出（Buffer Overflow）问题，这种问题主要出现在主要出现在 C/C++ 语言编写的系统中，它造成的漏洞是当今绝大多数安全漏洞的主要根源。在 Java / .NET 等利用虚拟机的(托管)平台上不会产生此问题。

要避免此问题，则必须对系统输入数据进行严格的长度检查，废弃或截断超长的越界数据，同时利用基础库函数中的一些更为安全的字符串处理函数来处理数据，也可以利用编译器或代码复查工具提供的检查功能来尽早发现可能会产生问题的程序。

1.2.2 输入非法数据

恶意的攻击者会尝试在用户界面或接口中向系统输入恶意数据，以便期望绕过系统的安全限制，致使系统出甚至崩溃或其他非法目的，因此在编码时，须要对所有输入数据（包括用户在界面中输入的数据和其他应用系统通过接口传递的数据）进行严格的合法性检查。

1.2.3 SQL 注入式攻击

SQL 注入式（SQL Injection）攻击是一种典型的，因对输入数据不当处理而产生的非常严重的安全漏洞。其原因是基于数据库的应用程序中经常会使用动态 SQL 语句，而且在程序又没有对输入数据严格检查，致使攻击者能在界面层或接口层注入非法的 SQL 语句，从而非法访问和破坏数据、反向工程、甚至对服务器本身造成

威胁。对于攻击者来说，SQL注入式攻击是一种简单有效的攻击方式，也是首选方式，尤其是在基于 Web的应用程序中，因此开发人员必须重点关注此问题。

预防 SQL注入式攻击的手段就是严格检查用户输入的数据，要使用基础系统提供的参数化查询接口，避免使用字符串来构造动态 SQL查询。同时对于数据库对象的访问权限进行严格限制，避免恶意 SQL语句破坏数据或系统。

1.2.4 拒绝服务攻击

拒绝服务攻击（Denial of Services -DoS）是指通过大量并发访问，使得服务器的有限特定资源（如网络、处理器、内存等）接近枯竭，使得服务器或操作系统失效的攻击行为。

DoS攻击的一般方式有发送大量数据包造成网络阻塞、执行内存泄漏代码使得系统可用内存越来越少、执行大量消耗 CPU处理能力的代码、通过客户端发送大量的 HTTP请求造成巨量 Web点击以及 SYN Flood等。DoS 攻击虽然不会直接对服务器本身带来损坏，但它使得真正的合法用户无法访问系统，从而可能带来业务上的损失。除了 DoS之外，攻击者还可能利用数量庞大的攻击源发起 DDoS(Distributed DoS，分布式拒绝服务）攻击，其破坏和危害作用更大。

在编码时要注意防范可能的 DoS攻击，具体措施包括提高软件行为的可管理性、主动拒绝异常连接、自动锁定攻击源、提供实时监控界面，能够有效甄别攻击源、具有(异常)事件报警机制、具有审核日志等。通过这些主动或被动的防御手段，能够将 DoS/DDoS攻击行为带来的破坏和危害降到较低水平。

1.2.5 敏感信息泄露

攻击者可能会通过暴力攻击、侦听、截取中间数据、反向工程、社会工程学（Social Engineering）等手段，获取访问凭据或机密信息，危及数据的私有性/安全性或者暴露敏感的商业数据，如用户名/口令、加密密钥、数据库连接串、商业敏感信息等。

因此在处理这些数据时，必须利用以密码技术为主的安全技术来进行强有力的机密性保护。在使用密码技术时，一般要利用公开的、经过广泛验证的可靠加密算法，同时加强密钥的管理和保护。

（签字页）

发布时间	2020-08-16
评审意见	无已披露补充评估
经办人签字 （甲方）	2020.12.16
乙方单位签章	

网络安全与保密协议

甲方：重庆中烟工业有限责任公司

乙方：东华软件股份公司

根据《网络安全法》规定，本着”诚实守信“原则，经甲、乙双方协商一致，自愿签订本协议，以资共同遵守。

第一条 知识产权

甲方向乙方购买的任何产品及服务，包括硬件设备、成品软件系统、定制开发软件，以及系统收集、产生的数据和相关信息的所有权、使用权均属于甲方。

第二条 保密信息

2.1 甲方购买乙方任何定制开发软件系统的源代码以及开发调试过程中使用、产生的相关数据、信息。

2.2 甲方购买乙方任何硬件设备的在安装调试过程中使用、产生的相关数据、信息。

2.3 甲方购买乙方任何软件系统（包含定制开发软件系统）在运行过程中收集、产生的数据和相关信息。

2.4 甲方使用的硬件、软件系统的体系架构、技术参数、合同价格等信息，以及硬件、软件系统运行过程中收集、产生的数据和相关信息。

2.5 甲乙双方共同履行针对合作意图、技术路径和管理要求等相关内容。

2.6 甲方内部组织机构信息、业务信息、产品信息等有关甲方运营生产相关信息。

2.7 以上保密信息任何方式存储的各种媒介产品，包括但不限于文档、光盘、存储设备等。

第三条 双方权利和义务

3.1 乙方保证保密信息仅用于提供给甲方服务工作，乙方不得将保密信息用于其他用途。除此之外，乙方不得对保密信息擅自读取、存储、缓存、使用和转接、转交第三方。任何情况下，乙方不得对外发布保密信息。

3.2 乙方保证对保密信息予以妥善保管，若发生以下事项由乙方承担全部责任：

3.2.1 保密信息因乙方原因被盗、泄露，或者以其他方式泄露、损毁、灭失。

3.2.2 乙方任何有权获得保密信息的人员对保密信息未经甲方授权的披露。

3.3 乙方保证对甲方提供的保密信息予以保密，不得用于其他商业用途。

3.4 乙方在工作完成后应及时将承载保密信息的介质原件及复制件全部销毁。

3.5 乙方提供给甲方的软件、硬件产品安全和信息系统安全必须符合国家法律及行业要求。

3.6 甲乙双方合作的具体业务应遵循行业有关要求并接受国家局、总公司有关部门和专业公司工作指导。

3.7 乙方得知第三方获得甲方任何保密信息，应第一时间通知甲方，并向甲方提供掌握的所有相关情况。

第四条 违约责任

乙方未履行或未完全履行本协议的条款均构成违约，违约方承担因此造成的一切保密、经济、法律责任，同时赔偿甲方的一切损失，包括但不限于甲方调查违约行为而支付的合理费用。

第五条 法律适用和争议解决

本协议适用中华人民共和国法律。所有因本协议引起的或与本协议有关的任何争议通过双方友好协商解决。如果协商不成，任何一方均可通过法律途径保障自己的合法权限。

第六条 一般条款

6.1 协议任何条款的无效不影响本协议其他条款的有效性。

6.2 本协议自双方签字盖章之日起生效。有效期至乙方软件、硬件废弃之日止。

甲方：

乙方：