信息安全相关鉴定项

中车长春轨道客车股份有限公司

2019-08-24

文档修订

版本	日期	修改人员	描述	审核人员
1.0	2019-08-24	陈仁合	创建	胡锐

一. 摘要

漏洞类型检测项目如下：

检测分类	检测项
Web 安全	SQL 注入
	跨站脚本攻击（XSS）
	XML 外部实体（XXE）注入
	跨站点伪造请求（CSRF）
	服务器端请求伪造（SSRF）
	任意文件上传
	任意文件下载或读取
	任意目录遍历
	.svn/.git 源代码泄露
	信息泄露
	CRLF 注入
	命令执行注入
	URL 重定向
	Json 劫持
	第三方组件安全
	本地/远程文件包含
	任意代码执行
	Struts2 远程命令执行
	Spring 远程命令执行
	缺少“X-XSS-Protection”头
	flash 跨域
	HTML 表单无 CSRF 保护
	HTTP 明文传输
	使用 GET 方式进行用户名密码传输
	X-Frame-Options Header 未配置

检测分类	检测项
	任意文件删除
	绝对路径泄露
	未设置 HTTPONLY
	X-Forwarded-For 伪造
	明文传输
	不安全的 HTTP Methods
	任意文件探测
网络传输安全	加密方式不安全
	使用不安全的 telnet 协议
业务逻辑安全	验证码缺陷
	反序列化命令执行
	用户名枚举
	用户密码枚举
	用户弱口令
	会话标志固定攻击
	平行越权访问
	垂直越权访问
	未授权访问
	业务逻辑漏洞
	短信炸弹
	Flash 未混淆导致反编译
中间件安全	中间件配置缺陷
	中间件弱口令
	Jboss 反序列化命令执行
	Websphere 反序列化命令执行
	Jenkins 反序列命令执行
	JBoss 远程代码执行
	Webloigc 反序列化命令执行
	Apache Tomcat 样例目录 session 操纵
服务器安全	文件解析代码执行

检测分类	检测项
	域传送漏洞
	Redis 未授权访问
	MongoDB 未授权访问
	操作系统弱口令
	数据库弱口令
	本地权限提升
	已存在的脚本木马
	永恒之蓝
	mssql 信息探测
	windows 操作系统漏洞
	数据库远程连接
	权限分配不合理

二. 各类检测项概述

2.1 Web 安全

2.1.1 SQL 注入

漏洞概述

SQL 注入攻击主要是由于程序员在开发过程中没有对客户端所传输到服务器端的参数进行严格的安全检查，同时 SQL 语句的执行引用了该参数，并且 SQL 语句采用字符串拼接的方式执行时，攻击者将可能在参数中插入恶意的 SQL 查询语句，导致服务器执行了该恶意 SQL 语句。SQL 注入漏洞主要影响是攻击者可利用该漏洞窃取数据库中的任意内容，在某些场景下，攻击者将有可能获得数据库服务器的完全控制权限。

2.1.2 跨站脚本攻击（XSS）

漏洞概述

跨站脚本攻击(Cross Site Scripting)，为不和层叠样式表(Cascading Style Sheets, CSS)的缩写混淆，故将跨站脚本攻击缩写为 XSS。恶意攻击者往 Web 页面里插入恶意 Script 代码，当用户浏览该页之时，嵌入其中 Web 里面的 Script 代码会被执行，从而达到恶意攻击用户的目的。在不同场景下，XSS 有相应不同的表现形式，主要分为反射型、存储型以及 DOM 型的跨站脚本攻击，所造成的影响主要是窃取用户登录凭证（Cookies）、挂马攻击、页面访问挟持等。

2.1.3 XML 外部实体（XXE）注入

漏洞概述

XXE Injection 即 XML External Entity Injection,也就是 XML 外部实体注入攻击。漏洞是在对非安全的外部实体数据进行行处理时引发的安全问题。在 XML1.0 标准里里,XML 文档结构里定义了实体(entity)这个概念，实体可以通过预定义在文档中调用,实体的标识符可访问本地或远程内容。如果在这个过程中引入了“污染”源,在对 XML 文档处理后则可能导致信息泄漏、文件读取、命令执行等安全问题。

2.1.4 跨站点伪造请求（CSRF）

漏洞概述

CSRF（Cross-Site Request Forgery，跨站点伪造请求）是一种网络攻击方式，该攻击可以在用户毫不知情的情况下以用户自身的名义伪造请求发送给受攻击站点，从而在未授权的情况下执行在权限保护之下的操作。具体来讲，可以这样理解 CSRF 攻击：攻击者盗用了你的身份，以你的名义发送恶意请求，对服务器来说这个请求是完全合法的，但是却完成了攻击者所期望的一个操作，比如以你的名义发送邮件、发消息，盗取你的账号，添加系统管理员，甚至于购买商品、虚拟货币转账等。

2.1.5 服务器端请求伪造（SSRF）

漏洞概述

SSRF(Server-Side Request Forgery:服务器端请求伪造) 是一种由攻击者构造形成并由服务端发起请求的一个安全漏洞。一般情况下，SSRF 攻击的目标是从外网无法访问的内部系统。（正是因为它是由服务端发起的，所以它能够请求到与它相连而与外网隔离的内部系统）

SSRF 形成的原因大都是由于服务端提供了从其他服务器应用获取数据的功能且没有对目标地址做过滤与限制。比如从指定 URL 地址获取网页文本内容，加载指定地址的图片，下载等等。最终将可能导致，攻击者可通过外网服务器端利用该漏洞访问内网服务器端的资源。

2.1.6 任意文件上传

漏洞概述

任意文件上传漏洞主要是由于程序员在开发文件上传功能时，没有考虑对文件格式后缀的合法性进行校验或只考虑在应用前端（Web 浏览器端）通过 javascript 进行后缀校验，攻击者可上传一个包含恶意代码的动态脚本（如 jsp、asp、php、aspx 文件后缀）到服务器上，攻击者访问该脚本时服务器将对包含恶意代码的动态脚本解析，最终执行相应的恶意代码。该漏洞最终将可能直接影响应用系统的服务器安全，攻击者可通过所上传的脚本完全控制服务器。

2.1.7 任意文件下载或读取

漏洞概述

任意文件下载或读取漏洞主要是由于应用系统在提供文件下载或读取功能时，在文件路径参数中直接指定文件路径的同时并没有对文件路径的合法性进行校验，导致攻击者可通过目录跳转(..\或../)的方式下载或读取到原始指定路径之外的文件。攻击者最终可通过该漏洞下载或读取系统上的任意文件，如数据库文件、应用系统源代码、密码配置信息等重要敏感信息，造成系统的敏感信息泄露。

2.1.8 任意目录遍历

漏洞概述

任意目录遍历主要原因是由于应用系统所提供的目录浏览或文件浏览功能中，在处理当前路径参数时没有对参数进行合法性校验，攻击者可通过目录跳转的方式（../或..）浏览预想之外的目录信息。攻击者将可能利用该漏洞访问应用系统的任意文件目录，导致可浏览敏感目录下的文件信息，造成敏感信息泄露。

2.1.9 .svn/.git 源代码泄露

漏洞概述

.svn/.git 源代码泄露主要原因是由于应用系统开发人员或运维管理人员在对应用系统进行版本迭代更新时，没有及时对代码版本维护程序(svn 或 git)中所产生的代码索引或代码库文件进行及时清理，攻击者可通过读取该代码索引或代码库文件访问并下载应用系统的源代码信息，最终导致应用系统的源代码信息遭到泄露，攻击者可进一步通过源代码审计的方式挖掘应用系统中存在的安全隐患。

2.1.10 信息泄露

漏洞概述

信息泄露主要是由于开发人员或运维管理人员的疏忽所导致。如未及时删除调试页面、未关闭程序调试功能、未屏蔽程序错误信息、备份文件未删除、数据库备份文件未删除、未屏蔽敏感数据信息等多个方面所导致的不同严重程度的信息泄露。攻击者可通过所掌握的信

息进一步分析攻击目标，从而有效发起下一步的有效攻击。

2.1.11 CRLF 注入

漏洞概述

CRLF 注入即“HTTP 响应头拆分漏洞”，主要是由于应用系统在接收用户浏览器发送的参数信息后，参数信息在 HTTP 响应头中进行了输出并未经过有效的校验，攻击者可提交恶意的参数信息(\r\n)从而对 HTTP 响应头进行控制。攻击者可通过该漏洞发起 web 缓存感染、用户信息涂改、窃取敏感用户页面、跨站脚本漏洞等攻击，从而造成普通用户遭受到恶意攻击。

2.1.12 命令执行注入

漏洞概述

命令执行注入主要是由于开发人员在处理应用系统发起操作系统命令时引用了客户端参数，同时没有对该参数进行合法性校验，攻击者可在参数中注入恶意的命令参数，致使执行命令的过程中执行了攻击者所指定的恶意命令。通过该漏洞攻击者可执行任意的操作系统命令，在权限配置不当的情况下可直接获得操作系统的完全控制权限。

2.1.13 URL 重定向

漏洞概述

URL 重定向主要是由于 Web 应用系统在处理 URL 重定向时没有对接收到的 URL 参数进行合法性校验，攻击者可指定 URL 路径为恶意 URL 或恶意域名，当用户访问该 URL 重定向页面时将可能跳转到攻击者所指定的恶意页面，从而造成用户遭受到恶意代码的攻击。

2.1.14 Json 劫持

漏洞概述

Json 劫持主要是由于网站页面在响应用户请求时采用 Json 数组的方式进行返回，而该页面没有进行相应的合法判断，攻击者可在恶意网站上构造访问该页面的 URL 并诱惑用户进行点击访问，最终攻击者可通过 Javscript Hook 的方式窃取用户在该页面上所返回的 Json

数组信息，从而造成用户的敏感信息泄露。

2.1.15 第三方组件安全

漏洞概述

第三方组件主要包括 Ewebeditor、FCKeditor、Ueditor、JQuery 等常用第三方组件，开发人员在开发过程中调用第三方组件时并未考虑组件当前的安全状况，攻击者可通过第三方组件上的安全漏洞攻击应用系统，从而影响应用系统自身的安全。

2.1.16 本地/远程文件包含

漏洞概述

本地/远程文件包含漏洞主要出现在采用 PHP 开发的应用系统中，在 PHP 代码开发过程中较常采用文件包含的方式进行对代码的引用从而提高编码效率以及代码扩展性，被包含的文件内容将被当作 php 代码进行解析。当所需要包含的文件路径通过客户端浏览器进行提交时，攻击者可指定文件路径到本地或远程的恶意代码文件，应用系统将执行该文件中的恶意代码，最终攻击者可通过该方式获取应用系统的控制权限。

2.1.17 任意代码执行

漏洞概述

任意代码执行主要是由于应用系统在处理动态代码执行的过程中，部分代码片段可由客户端浏览器提交参数到服务器进行指定，从而攻击者可通过提交恶意的代码参数到服务器，应用系统将执行所提交的恶意代码，最终攻击者可通过该漏洞直接获得应用系统的控制权限。

2.1.18 Struts2 远程命令执行

漏洞概述

Struts2 远程命令执行主要是由于网站采用较低版本的 Strut2 框架，该框架低版本在处理远程客户端参数名、参数值、文件名参数等参数内容时没有经过严格的过滤，导致可注入到 OGNL 表达式中，从而造成任意代码执行漏洞。攻击者可通过构造恶意的 OGNL 表达式从而实现任意命令执行，最终可通过该漏洞完全获得网站权限甚至操作系统权限。

2.1.19 Spring 远程命令执行

漏洞概述

Spring 远程命令执行主要是由于网站采用较低版本的 Spring 框架，该框架低版本在处理 Spring 标签时没有进行合法性校验，导致可将标签内容信息注入到表达式中，从而造成任意代码执行漏洞。攻击者可通过构造恶意的标签内容到表达式从而实现任意命令执行，最终可通过该漏洞完全获得网站权限甚至操作系统权限。

2.1.20 缺少“X-XSS-Protection”头

漏洞概述

“X-XSS-Protection”头强制将跨站点脚本编制过滤器加入“启用”方式，即使用户已禁用时也是如此。该过滤器被构建到最新的 web 浏览器中（IE 8+，Chrome 4+），通常在缺省情况下已启用。虽然它并非设计为第一个选择而且仅能防御跨站点脚本编制，但它充当额外的保护层。

2.1.21 flash 跨域

漏洞概述

flash 访问另一个域的数据，flash player 会自动从改域加载策略文件（crossdomain.xml）,如果访问的数据所在的域在策略文件中，则数据将可访问。

当 allow-access-from domain 字段值设置为*时，允许所有域名进行访问，就造成了 flash 跨域访问漏洞。

2.1.22 HTML 表单无 CSRF 保护

漏洞概述

跨站点请求伪造（缩写为 CSRF 或 XSRF）是一种恶意利用网站的方式，从网站信任的用户传输未经授权的命令。

攻击者可能会使用未被保护的 html 表单强制 Web 应用程序的用户执行攻击者选择的操

作。 成功的 CSRF 攻击可以在普通用户的情况下危害目标用户数据和操作。 如果管理员帐户被 CSRF 攻击，有可能会危及整个 Web 应用程序。

2.1.23 HTTP 明文传输

漏洞概述

在测试过程中，检测到基于不安全的 http 连接的登录请求。由于采用未加密的 http 请求发送敏感的登录请求，有可能被同一个局域网内的攻击者嗅探到用户输入的登录数据，如账号和密码。

2.1.24 使用 GET 方式进行用户名密码传输

漏洞概述

因为 GET 方式传输请求的数据会被浏览器缓存起来，用户名和密码将明文出现在 URL 上，其他人可以通过查询历史浏览记录来查询密码。

2.1.25 X-Frame-Options Header 未配置

漏洞概述

攻击者可以使用一个透明的、不可见的 iframe，覆盖在目标网页上，然后诱使用户在该网页上进行操作，此时用户将在不知情的情况下点击透明的 iframe 页面。通过调整 iframe 页面的位置，可以诱使用户恰好点击 iframe 页面的一些功能性按钮上，导致被劫持

X-Frame-Options HTTP 响应头可以指示浏览器是否允许当前网页在“frame”或“iframe”标签中显示，以此使网站内容不被其他站点引用和免于点击劫持攻击。

利用该漏洞可能导致点击劫持攻击，用来钓鱼，具体危害看具体情况。

2.1.26 任意文件删除

漏洞概述

任意文件删除漏洞主要是由于应用系统在提供文件删除功能时，在文件路径参数中直接

指定文件路径的同时并没有对文件路径的合法性进行校验，导致攻击者可通过目录跳转(..\或../)的方式删除原始指定路径之外的文件。攻击者最终可通过该漏洞删除系统上的任意文件，如数据库文件、应用系统源代码、密码配置信息等重要敏感信息，造成系统的敏感信息泄露。

2.1.27 绝对路径泄露

漏洞概述

系统存在绝对路径泄露，绝对路径是指目录下的绝对位置，直接到达目标位置，通常是从盘符开始的路径。攻击者可通过此漏洞了解到整个服务器的目录结构，同时如果存在注入漏洞，攻击者可通过绝对路劲直接获得服务器权限。

2.1.28 未设置 HTTPONLY

漏洞概述

网站 Cookie 特别是用于用户身份认证的 JSESSIONID 未设置 HttpOnly 选项，当网站存在跨站脚本漏洞时，攻击者可获取该 Cookie，并登陆受害者账号。

2.1.29 X-Forwarded-For 伪造

漏洞概述

可对 X-Forwarded-For 头信息进行伪造

2.1.30 明文传输

漏洞概述

在测试过程中，检测到基于不安全的传输请求。由于采用未加密的方式发送敏感的登录请求，有可能被同一个局域网内的攻击者嗅探到用户输入的登录数据，如账号和密码。

2.1.31 不安全的 HTTP Methods

漏洞概述

不安全的 HTTP 方法主要有 PUT/DELETE/MOVE/COPY/TRACE,通过此类扩展方法，可能上传文件到服务器或删除服务器上的文件。

2.1.32 任意文件探测

漏洞概述

可通过该漏洞判断服务器中某个文件是否存在

2.2 网络传输安全

2.2.1 加密方式不安全

漏洞概述

在数据传输过程或储存敏感信息时采用的加密算法不安全

2.2.2 使用不安全的 telnet 协议

漏洞概述

Telnet 本身存在缺陷：

1、没有口令保护，远程用户的登陆传送的帐号和密码都是明文，使用普通的 sniffer 都可以被截获

2、没有强力认证过程。只是验证连接者的帐户和密码。

3、没有完整性检查。传送的数据没有办法知道是否完整的，而不是被篡改过的数据。

4、传送的数据都没有加密。

2.3 业务逻辑安全

2.3.1 验证码缺陷

漏洞概述

常见于应用系统在登录处理流程过程中，当用户登录失败后并未对当前验证码进行注销并重新刷新验证码，攻击者可至始至终提交初始的验证码发起攻击穷举攻击；同时部分应用系统验证码只在客户端浏览器验证，并未经过服务器远程验证，将可能存在绕过验证码缺陷，另一方面，在生成或获取验证码的过程中存在缺陷，攻击者将可能成功预测验证码内容或直接解析验证码内容，从而使验证码失去原有意义，最终导致一系列的穷举或遍历数据攻击。

2.3.2 反序列化命令执行

漏洞概述

反序列化命令执行主要是由于应用系统在通过反序列的方式处理字节序列时没有对该序列信息进行校验，攻击者可伪造恶意的字节序列并提交到应用系统时，应用系统将对字节序列进行反序列处理时将执行攻击者所提交的恶意字节序列，从而导致任意代码或命令执行，最终可完成获得应用系统控制权限或操作系统权限。

2.3.3 用户名枚举

漏洞概述

在应用系统登录过程中，当输入错误的用户名信息时，应用程序将反馈相应的诸如“用户不存在”的错误提示，攻击者可通过该提示为依据进行对用户名的枚举，猜解出已存在于应用系统的用户名信息，最终攻击者可进行一步发起对已有用户的密码猜解。

2.3.4 用户密码枚举

漏洞概述

在应用系统登录过程中，由于并未限制用户的密码输入错误次数，攻击者可通过密码字典不断枚举指定用户的密码信息，最终用户密码将可能遭受到破解，攻击者可通过所破解的

用户密码进入应用系统并发起进一步的攻击。

2.3.5 用户弱口令

漏洞概述

由于应用系统的相关用户的安全意识薄弱，同时应用系统并未有效的强制性密码策略要求，从而将可能存在弱口令用户，攻击者可轻易通过字典猜解的方式获得用户的密码并进入应用系统发起进一步攻击。

2.3.6 会话标志固定攻击

漏洞概述

应用系统在用户登录成功或登录失败后并未对当前的会话标志（Session ID）进行更新，从而攻击者可构造一个未登录且带有 Session ID 的 URL 并发送到用户，用户点击该 URL 并进行登录成功后，攻击者可通过该 Session ID 冒充用户并成功进入应用系统，从而可进一步发起对应用系统的攻击。

2.3.7 平行越权访问

漏洞概述

应用系统在处理同一业务功能数据时，并未对数据与当前用户的权限进行合法性校验，从而导致用户可越权访问、篡改、删除、添加其他用户的信息，造成越权操作。常见如：访问任意用户订单、修改任意用户密码、删除任意用户信息等。

2.3.8 垂直越权访问

漏洞概述

应用系统在处理各个角色业务功能时，并未对当前用户角色与该业务功能的权限标志进行判断，导致用户可越权访问非自身权限范围内的业务功能，造成越权操作。常见如：越权添加、修改、删除用户以及权限、越权访问系统管理功能等。

2.3.9 未授权访问

漏洞概述

应用系统对业务功能页面并未进行有效的身份校验，在未登录且获知业务功能页面的访问地址前提下，可直接操作该页面下的功能，将可能对应用系统的恶意破坏。

2.3.10 业务逻辑漏洞

漏洞概述

此类问题涉及面较广，主要是由于开发阶段对于程序逻辑的设计和限制存在缺陷导致。

常见问题点：

- 在线支付
- 顺序执行
- 本地限制，抓包绕过

1.在线支付

1 支付过程中可直接修改数据包中的支付金额

此类漏洞是支付漏洞中最常见的。开发人员为了方便，在支付的关键步骤数据包中直接传递需要支付的金额。同时后端程序没有对金额做正确的校验，传递过程中也没有任何签名措施，导致可以随意篡改金额提交。

1 没有对购买数量进行限制

产生的原因是开发人员没有对购买的数量参数进行严格的限制。这种同样是数量的参数没有做签名，导致可随意修改，经典的修改方式就是改成负数。当购买的数量是一个负数时，总额的算法仍然是“购买数量 x 单价 = 总价”，会导致产生一个负数的需支付金额。若支付成功，则可能导致购买到了一个负数数量的产品，并有可能返还相应的积分/金币到攻击者账户中。

类似的，也可以将数量改成一个较大的值，导致数值溢出，交易订单出现问题。

1 请求重放

未对订单唯一性进行验证，导致购买商品成功后，重放其中请求，可以使购买商品一直增加。

1 其他参数干扰

由于对商品价格，数量等以外的其它会影响最终金额参数(如：运费)缺乏验证导致最终金额可被控制。

2.顺序执行

1 密码找回的顺序执行

正常流程应为：填写用户名，并向指定的邮箱或手机号发送校验信息，根据邮箱的 URL 或手机短信进行验证后进入重置密码界面，最后成功修改重置密码。

整个环节中，如果对前提过程验证不够严格，就会导致前续操作可被绕过，从而导致任意用户密码重置漏洞。、

3.本地限制 重放攻击

通常情况下我们需要对用户在网页中的各种操作及输入进行限制，以促使用户的输入符合预期。如限制用户输入邮箱地址，手机号码，限制用户上传的文件类型，要求用户输入正确的验证码等等。

为了实现这个目的， 。但是部分开发人员过分依赖和相信在前端插入 JavaScript 脚本的方法，忽视了在后端对用户输入的处理，导致漏洞。

2.3.11 短信炸弹

漏洞概述

业务端口安全防范措施落实不到位，未对验证码短信下发总次数进行限制，且发送间隔设置不合理，导致被不法分子恶意调用，用于“短信炸弹”等非法目的，向正常用户无节制发送验证码短信或消耗服务器资源。

2.3.12 Flash 未混淆导致反编译

漏洞概述

Flash 未混淆导致反编译

2.4 中间件安全

2.4.1 中间件配置缺陷

漏洞概述

中间件配置缺陷主要是由于开发人员或运维人员在安装部署中间件后，并未对默认的中间件配置进行安全加固，导致产生一系列诸如目录遍历、默认示例文件、错误信息泄露等缺陷，从而导致应用系统产生信息泄露隐患。

2.4.2 中间件弱口令

漏洞概述

中间件弱口令主要是由于开发人员或运维人员在部署中间件过程中，并未对中间件控制台进行口令配置或修改默认口令，从而攻击者可通过穷举猜解的方式进行中间件控制台，最终攻击者可通过控制台上传恶意脚本并获得应用系统权限。

2.4.3 Jboss 反序列化命令执行

漏洞概述

由于 Jboss 版本并未进行及时更新，在低版本中的 Commoncollections.jar 程序包中存在反序列化命令执行漏洞，攻击者可远程构造恶意的字节序列发送到服务器端并执行，通过该漏洞攻击者可直接获得应用系统控制权限甚至操作系统权限。

2.4.4 Websphere 反序列化命令执行

漏洞概述

由于 Websphere 版本并未进行及时更新，在低版本中的 Commoncollections.jar 程序包中存在反序列化命令执行漏洞，攻击者可远程构造恶意的字节序列发送到服务器端并执行，通过该漏洞攻击者可直接获得应用系统控制权限甚至操作系统权限。

2.4.5 Jenkins 反序列命令执行

漏洞概述

由于 Jenkins 版本并未进行及时更新，在低版本中的 Commoncollections.jar 程序包中存在反序列化命令执行漏洞，攻击者可远程构造恶意的字节序列发送到服务器端并执行，通过该漏洞攻击者可直接获得应用系统控制权限甚至操作系统权限。

2.4.6 JBoss 远程代码执行

漏洞概述

由于 Jboss 版本未进行及时更新，在低版本中存在未授权访问漏洞从而可直接访问 EJBInvokerServlet 或 JMXInvokerServlet 功能模块，通过该功能模块攻击者可远程部署恶意的 WAR 应用程序包，最终攻击者可成功部署恶意代码到应用系统服务器上，从而获得应用系统权限。

2.4.7 Webloigc 反序列化命令执行

漏洞概述

由于 Weblogic 版本并未进行及时更新，在低版本中的 Commoncollections.jar 程序包中存在反序列化命令执行漏洞，攻击者可远程构造恶意的字节序列发送到服务器端并执行，通过该漏洞攻击者可直接获得应用系统控制权限甚至操作系统权限。

2.4.8 Apache Tomcat 样例目录 session 操纵

漏洞概述

Apache Tomcat 默认安装包含”/examples”目录，里面存着众多的样例，其中 session 样例(/examples/servlets/servlet/SessionExample)允许用户对 session 进行操纵。因为 session 是全局通用的，所以用户可以通过操纵 session 获取管理员权限。

2.5 服务器安全

2.5.1 文件解析代码执行

漏洞概述

由于网站所采用的中间件版本过低，不同的低版本中间件均存在文件解析执行漏洞，攻击者可直接上传包含恶意代码的图片文件或压缩文件到网站服务器上，该图片或压缩文件将以动态脚本代码的方式进行解析并执行，最终攻击者可通过执行恶意代码获得网站的控制权限。

2.5.2 域传送漏洞

漏洞概述

域传送漏洞主要由于 DNS 服务器开启了域传送功能，同时并未采用有效的白名单机制指定域传送的传送范围，导致攻击者可远程获得 DNS 服务器上的 DNS 记录，造成网站的域名以及 IP 地址信息的泄露。

2.5.3 Redis 未授权访问

漏洞概述

Redis 未授权访问主要是由于默认安装的情况下并未对 Redis 配置身份校验功能，导致攻击者可非法访问 Redis 下的数据信息，同时可进一步通过配置文件功能对服务器文件进行写入，如写入 SSH 密钥或计划任务，从而获得服务器的控制权限。

2.5.4 MongoDB 未授权访问

漏洞概述

MongoDB 未授权访问主要是由于默认安装的情况下并未对 MongoDB 配置身份校验功能，导致攻击者可非法访问 MongoDB 下的数据信息，从而造成 MongoDB 中的数据库信息泄露。

2.5.5 操作系统弱口令

漏洞概述

操作系统弱口令主要由于运维管理员的安全意识薄弱，对管理员账号设置了简单密码，攻击者可通过字典穷举的方式破解管理员密码并完全获得服务器控制权限。

2.5.6 数据库弱口令

漏洞概述

数据库弱口令主要由于运维管理员的安全意识薄弱，对数据库管理员账号设置了简单密码，攻击者可通过字典穷举的方式破解管理员密码并完全获得数据库控制权限。在特定场景下，攻击者可通过数据库权限进行权限提升，从而进一步获得数据库所在服务器的控制权限。

2.5.7 本地权限提升

漏洞概述

本地权限提升漏洞主要是由于服务器没有及时更新操作系统补丁、数据库补丁或其他第三方应用程序补丁，导致攻击者可利用存在的安全缺陷或配置缺陷进行普通权限到最高权限的权限提升，最终可直接获得服务器的最高权限。

2.5.8 已存在的脚本木马

漏洞概述

在渗透测试过程中发现非渗透测试人员所上传的脚本木马，主要是攻击者此前对应用系统发起了攻击并成功上传脚本木马并获得了相应的权限。

2.5.9 永恒之蓝

漏洞概述

EternalBlue（永恒之蓝）据称是方程式组织在其漏洞利用框架中一个针对 SMB 服务进行攻击的模块，由于其涉及漏洞的影响广泛性及利用稳定性，在被公开以后为破坏性巨大的勒索蠕虫 WannaCry 所用而名噪一时。通过未打补丁主机漏洞，可远程控制执行任意代码。

2.5.10 mssql 信息探测

漏洞概述

SQL Server 服务使用两个端口：TCP-1433、UDP-1434。其中 1433 用于供 SQL Server 对外提供服务，1434 用于向请求者返回 SQL Server 使用了哪个 TCP/IP 端口及版本信息。

2.5.11 windows 操作系统漏洞

漏洞概述

windows 操作系统未及时更新补丁，存在可被利用的漏洞，攻击者可在远程攻击致使服务器宕机或直接获取操作系统权限

2.5.12 数据库远程连接

漏洞概述

数据库管理员为方便管理、配置数据库，开启数据库远程连接功能。攻击者可暴力枚举数据库用户名及密码进行登陆尝试，威胁数据库及服务器安全。

2.5.13 权限分配不合理

漏洞概述

系统服务分配给用户的权限过高，导致普通用户可以对服务器执行高危的操作，比如执

行系统命令，获取系统敏感信息等。

注： 该文档最终解释权为中车长春轨道客车股份有限公司所有

该文档将随着信息化的发展逐渐动态更新