|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目名称 | 业务系统运维服务项目-安全等保常态化服务项目 | | | | |
| 项目编号 |  | | 项目经理 |  | |
| 密　　级 | □公开使用 ■内部受控 | | | | |
| 总 页 数 |  | 正　　文 |  | 附　　件 |  |
| 编制/日期 | 2024年12月12日 | | 审核/日期 | 2024年12月12日 | |
| 批准/日期 |  | | | | |

**北京市生态环境局**

**重点排污单位自动监控与基础数据库系统**

**深度扫描测试报告**

**北京华夏航通信息技术有限公司**

**2024年12月12日**

**文件修订记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本号** | **变化状态** | **简要说明** | **变更人** | **变更日期** | **备注** |
| V1.0 | C | 创建编制 | 深度扫描技术 | 20241212 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

变化状态：C-创建，A-增加，M-修改，D-删除

|  |  |  |  |
| --- | --- | --- | --- |
| **测试对象** | 重点排污单位自动监控与基础数据库系统 | | |
| **测试类别** | 黑盒测试 | **委托日期** | 2024年12月12日 |
| **检测时间** | 2024年12月12日~ 2024年12月12日 | | |
| **测试结果** | 本次为第1次深度扫描测试。  通过本次深度扫描测试工作，在深度扫描测试范围内共发现2个可能潜在的安全问题。安全问题按照风险等级从高到低可分为高、中、低三个等级，其中高风险级别的问题有2个，中风险级别的问题0个，低风险为0个。  详情请看“安全漏洞详细分析”。  2024年12月12日 | | |

# 项目概述

## 前言

Web应用是网络应用和业务的集成，为所有用户提供方便的信息共享和应用共享。Web业务平台已经在电子商务、企业信息化中得到广泛的应用，企业都纷纷将应用架设在Web平台上，为客户提供更为方便、快捷的服务支持。伴随着这种趋势，入侵者也将注意力从以往对传统网络服务器的攻击逐步转移到了对 Web 业务的攻击上。根据 Gartner 的调查，信息安全攻击有 75% 都是发生在 Web 应用层而非网络层面上。同时，数据也显示，三分之二的 Web 站点都相当脆弱和易受攻击。

为了保障Web应用的稳定、安全、持续运行，就要对网上业务系统的信息安全风险做到早发现、早预防。深度扫描测试将结合国际国内信息安全最佳安全实践，通过对外网业务系统进行深度扫描测试，并根据测试结果提出相应的整改加固建议，协助用户完成整改加固实施，最终提高用户网上业务系统的安全水平。

## 深度扫描测试目的

分析客户WEB应用系统的安全现状，检测WEB应用系统的漏洞和安全问题，并验证其他已知的脆弱点。对系统的任何弱点、技术缺陷或漏洞的主动分析，并且以有利于攻击为目的而对漏洞加以利用。从而全面了解和掌握WEB应用系统的信息安全威胁和风险，为WEB应用系统开展安全调优及加固建设提供依据，并指导客户实施调优及加固工作，具体的目标包括：

帮助客户理解应用系统当前的安全状况，发现在系统复杂结构中的最脆弱链路；

通过改进建议，保证WEB应用系统和相关基础设施满足标准的安全性基线；

降低WEB应用系统信息安全事件发生的可能性；

保障WEB应用系统的安全、可靠、稳定运行。

## 深度扫描测试组织

为了保证深度扫描测试工作的顺利实施，确保项目质量达到预期目标，加强管理和协调合作，使工作和责任更加清晰明确，特建立分工明确，职责清楚，层次分明同时又能协调配合的项目管理组织结构。

深度扫描测试实施按照工作内容划分为以下职能小组：

* 领导协调小组：
  + 沟通协调参与各方的相关事宜；
  + 监督、控制项目进度及质量，及时发现项目实施过程中存在的问题，并及时组织改进；
  + 对重大问题进行决策；
* 项目经理：
  + 负责项目各阶段和总体工作的准备和实施工作；
  + 总体把握项目风险控制、时间安排和项目质量工作；
  + 向项目领导协调小组反馈问题；
  + 组织项目验收工作。
* 深度扫描测试小组：
  + 负责深度扫描测试中的信息收集、深度扫描测试和报告编写等各项实施工作；
  + 向项目经理反馈问题。

## 深度扫描测试原则

本次深度扫描测试工作中严格遵循以下原则：

标准性原则：深度扫描性测试方案的设计和实施应依据行业、国家、国际的相关标准进行；

规范性原则：服务提供商的工作过程和所有文档，应具有很好的规范性，以便于项目的跟踪和控制；

可控性原则：在保证测试质量的前提下，按计划进度执行，需要保证对深度扫描测试工作的可控性。安全评估的工具、方法和过程要在双方认可的范围之内合法进行；

整体性及有限性原则：深度扫描性测试的内容应包括用户等各个层面，深度扫描性测试的对象应包括和仅限于用户所指定的具体设备及系统，未经用户授权不得减小或扩大深度扫描性测试的范围和对象；

最小影响原则：深度扫描性测试工作应尽量避免影响系统和网络的正常运行，不对正常运行的系统和网络构成破坏和造成停产；

保密原则：深度扫描性测试的过程和结果应严格保密，不能泄露测试项目所涉及的任何打印和电子形式的有效数据和文件。

# 深度扫描测试方法



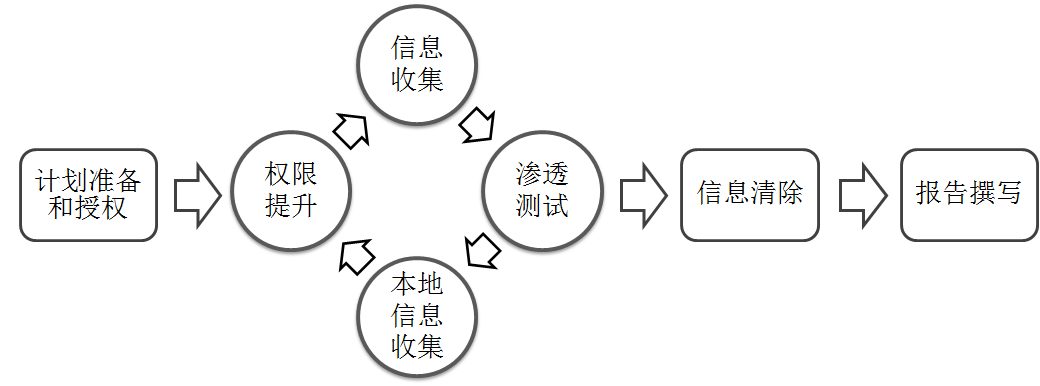
## 概述

深度扫描测试工作主要是对于已经采取了安全防护措施（安全产品、安全服务）或者即将采用安全防护措施的用户而言，明确网络当前的安全现状并对下一步的安全建设有重大的指导意义。深度扫描测试服务用于验证在当前的安全防护措施下网络、系统抵抗黑客攻击的能力。

任务组利用各种主流的攻击技术对网络、系统和应用进行远程攻击测试，目的是发现网络、系统和应用层面存在的安全漏洞和隐患，并提出相应的整改建议。

## 深度扫描测试方法

此次深度扫描测试的工作流程如下图所示：



计划和准备：在实际工作开展前需要同客户就深度扫描测试的目标、范围、工作方式以及实施成果进行讨论，并签订保密协议，成立深度扫描测试工作小组，在得到客户的深度扫描测试授权书后开展工作。

信息收集操作：收集深度扫描性测试所需信息（包括设备存活情况、网络拓扑情况以及扫描发现漏洞情况等），准备相关测试工具。

深度扫描测试：按照互联网深度扫描性测试实施计划进行测试验证。

本地信息收集：根据深度扫描深度情况，不断收集相关环境信息及漏洞情况，并进行综合分析。

权限提升：根据环境条件进行相关的权限提升，直到最高权限。

循环：在整个深度扫描过程中，就是不断地进行信息收集操作、深度扫描测试、本地信息收集和权限提升这一系列操作过程。

信息清除：根据深度扫描过程中使用的一些工具情况进行必要的清除过程。

报告撰写：根据深度扫描过程输出报告。

## 风险控制和管理

在深度扫描测试中，测试过程的安全可控是非常重要的。按照相关规定，任务组特别注意了测试过程本身的安全性问题，制定了安全控制措施和安全控制策略，采取了适当的风险规避、风险降低等方法，尽量避免影响业务的正常运行。在某些测试中，若无法确定测试过程是否安全，任务组将放弃相关测试以确保测试对象的安全。

另外，对于安全防护措施严密的网络、系统，在有限的时间内进行深度扫描测试可能不会获得成功结果。这从另一角度证明了网络、系统能够在一定程度上抵抗黑客的攻击。

## 风险等级说明

风险等级划分标准如下：

|  |  |
| --- | --- |
| **风险等级** | **等级划分依据** |
| 高风险 | 获得管理员权限，完全控制机器和应用系统；利用该漏洞能够实现文件上传建立目录，从中获取大量的数据库表结构信息或利用该漏洞发起的攻击直接引起应用系统服务器故障、响应异常或资金损失。 |
| 中风险 | 获取部分访问权限，尚未完全控制机器和应用系统；利用该漏洞能够实现提取应用系统上相关信息，下载文件，但暂时无法实现文件上传或者建立目录。利用该漏洞攻击间接影响应用系统运行或引起客户敏感信息外泄。 |
| 低风险 | 无法验证的可能存在风险以及需要依赖客户风险意识薄弱方可触发的风险。 |

# 深度扫描测试情况

## 深度扫描测试范围

根据前期的调研和了解，本次深度扫描测试工作范围如下：

| **序号** | **网站名称** | **地址** |
| --- | --- | --- |
| 1 | 重点排污单位自动监控与基础数据库系统 | https://zdjk.sthjj.beijing.gov.cn/amOnline/zdjk-company-base/login |

## 深度扫描测试记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **测试时间** | **漏洞情况** | **项目人** | **测试号** | **备注** |
| 1 | 20241212 | 高危：2 中危：0 低危：0 | 深度扫描技术 | P1.0 |  |

## 风险等级概况

通过本次深度扫描测试工作，在深度扫描测试范围内共发现2个可能潜在的安全问题。安全问题按照风险等级从高到低可分为高、中、低三个等级，其中高风险级别的问题有2个，中风险级别的问题有0个，低风险为0个。

按照风险等级分类如下图所示：

# 安全漏洞详细分析

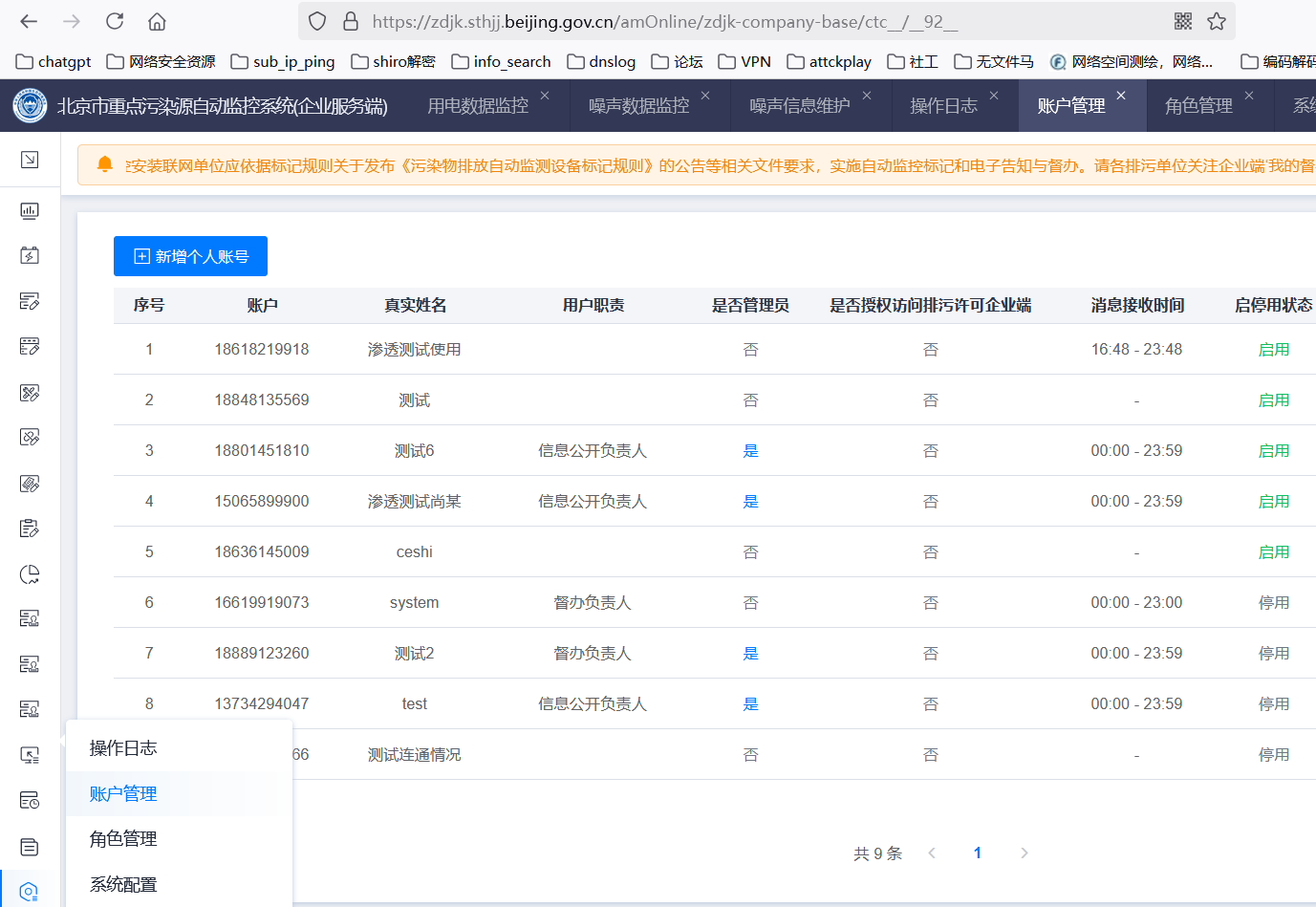
## 重点排污单位自动监控与基础数据库系统

### 垂直越权（高危）

#### 漏洞情况

使用测试用户登录系统后，访问系统管理--帐户管理，具有编辑权限，可将非管理员用户设置为管理员用户，也可进行新增个人账号，并设置为管理员权限。

首先将测试帐户进行编辑，修改为管理员。







并使用新增账号即可登录后台



#### 漏洞危害

黑客可以通过垂直越权获取服务器管理员权限，并通过相关权限进行进一步利用。

### 跨站脚本漏洞（高危）

#### 漏洞情况

POST /amOnline/app/monitorspot/spotAction!findAcceptInfoByID.page HTTP/1.1

Host: zdjk.sthjj.beijing.gov.cn

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:133.0) Gecko/20100101 Firefox/133.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded

Content-Length: 47

Origin: https://zdjk.sthjj.beijing.gov.cn

Connection: close

Referer: https://zdjk.sthjj.beijing.gov.cn/amOnline/app/monitorspot/spotAction!findAcceptInfoByID.page

Cookie: JSESSIONID=BFA4F951AAB4F92979423E3FCCB06090; jointframe.cluster.sessionid=HZ90DE6ECAAFB54C2FB5D24645CC150273; \_va\_id=830385d52861918e.1709972939.17.1733450237.1733450223.; \_va\_ref=%5B%22%22%2C%22%22%2C1733450223%2C%22https%3A%2F%2Fbjjp.bjysgl.cn%2F%22%5D; sensorsdata2015jssdkcross=%7B%22distinct\_id%22%3A%2218e5b602587a-01b0f0eb687cc4-e565623-1327104-18e5b602588822%22%7D; \_yfx\_session\_10018414=%7B%22\_yfx\_firsttime%22%3A%221733450230504%22%2C%22\_yfx\_lasttime%22%3A%221733450230504%22%2C%22\_yfx\_visittime%22%3A%221733450230504%22%2C%22\_yfx\_lastvisittime%22%3A%221733450230504%22%2C%22\_yfx\_domidgroup%22%3A%221733450230504%22%2C%22\_yfx\_domallsize%22%3A%22100%22%2C%22\_yfx\_cookie%22%3A%2220241206095710505876015016461890%22%7D; asToken=7ca0290c9a95d0268823e2b7706b354f; asEmplAccount=zhangzizhe2; focus-token=1005413243708b5f9b88375363bc03f86ef; focus-team-id=undefined; focus-app-id=undefined; asUserId=Cq6YcksN4ZB3nm0bsSFid

Upgrade-Insecure-Requests: 1

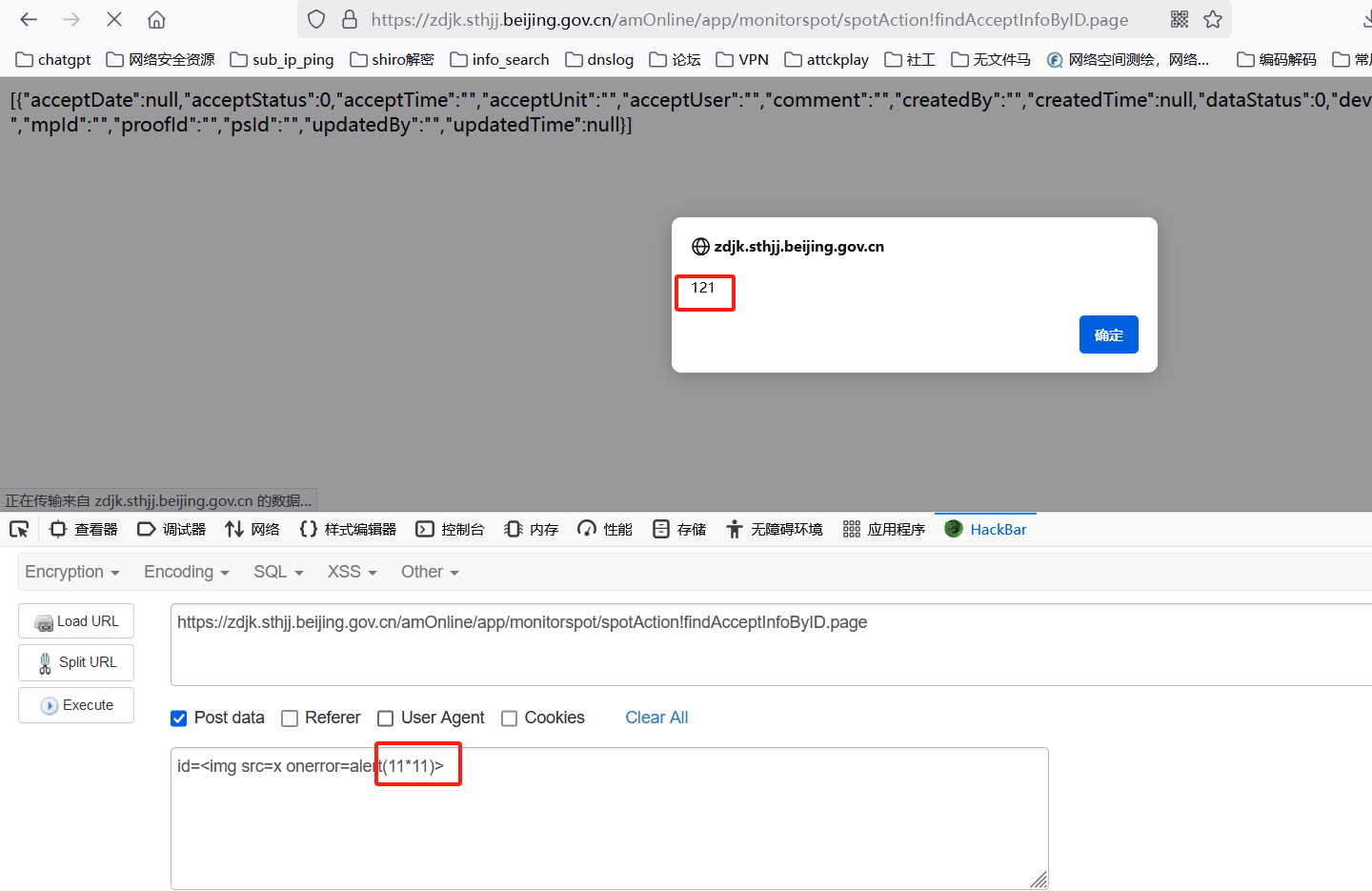
Sec-Fetch-Dest: document

Sec-Fetch-Mode: navigate

Sec-Fetch-Site: same-origin

Priority: u=0, i

id=%3Cimg+src%3Dx+onerror%3Dalert%2811\*11%29%3E



#### 漏洞危害

跨站脚本漏洞可以再分成两类：Stored attack 和 Reflected attack。这两种攻击的主要区别在于有效负荷到达服务器的方式。Stored attack 仅以某种形式存储在目标服务器上（例如在数据库中），或通过提交至公告板或访问者日志来进行存储。如果请求了所存储的信息，受害者将在自己的浏览器中检索和执行攻击代码。而 Reflected attack 则来自其他地方。当通过动态生成的网页中的服务器端，直接将 Web 客户端的用户输入包含在内时，就会出现 Reflected attack。通过某些社会工程手段，例如通过恶意链接或“伪装”表单，攻击者可以哄骗受害者提交信息，然后对信息进行修改以包含攻击代码，并将其发送至合法服务器。注入的代码随后将反射回用户的浏览器，由于此代码来自受信任的服务器，用户的浏览器将会执行此代码。两种攻击的影响都是相同的。

接口文档泄露

# 整改建议

整改方案建议是建立在对用户远程深度扫描测试结果的基础之上提出的整改建议。

## 针对越权访问整改建议

对最小化权限原则：在配置文件和系统设置中，遵循最小化权限原则，只给予必要的用户和进程最小限度的访问权限，减少敏感信息被意外泄露或滥用的风险。

## 针对跨站脚本漏洞整改建议

xss漏洞本质上是一种html注入，也就是将html代码注入到网页中。那么其防御的根本就是在将用户提交的代码显示到页面上时做好一系列的过滤与转义

1.假定所有输入都是可疑的，必须对所有输入中的script、iframe等字样进行严格的检查。这里的输入不仅仅是用户可以直接交互的输入接口，也包括HTTP请求中的Cookie中的变量，HTTP请求头部中的变量等。

2.不仅要验证数据的类型，还要验证其格式、长度、范围和内容。

3.不要仅仅在客户端做数据的验证与过滤，关键的过滤步骤在服务端进行。

4.对输出的数据也要检查，数据库里的值有可能会在一个大网站的多处都有输出，即使在输入做了编码等操作，在各处的输出点时也要进行安全检查。