**单位**：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2020年7月10日

CVE-2020-5902漏洞威胁分析报告

目录

[CVE-2020-5902漏洞威胁分析报告 1](E:\\WeChat Files\\WeChat Files\\wxid_0z4agx41rqny21\\FileStorage\\File\\2020-07\\CVE-2020-5902漏洞复现.docx" \l "_Toc45275577)

[1. 问题概述 3](#_Toc45275578)

[1.1. 什么是BIG-IP? 3](#_Toc45275580)

[1.2. 威胁程度 3](#_Toc45275581)

[2. 现状概述 4](#_Toc45275582)

[3. 威胁分析 4](#_Toc45275583)

[3.1. 攻击复现 5](#_Toc45275584)

[3.2. 漏洞原理解析 8](#_Toc45275585)

[3.3. 防御措施 9](#_Toc45275608)

[4. 总结 9](#_Toc45275609)

[5. 附件：北京市拥有相似漏洞的网站IP列表 9](#_Toc45275610)

**错误!超链接引用无效。错误!超链接引用无效。错误!超链接引用无效。错误!超链接引用无效。错误!超链接引用无效。错误!超链接引用无效。错误!超链接引用无效。错误!超链接引用无效。错误!超链接引用无效。错误!超链接引用无效。错误!超链接引用无效。错误!超链接引用无效。错误!超链接引用无效。**

# 问题概述

## 什么是BIG-IP?

BIG-IP ADC广泛应用于大型企业，数据中心、云计算平台中，可以实现应用加速、负载均衡、SLL过载、web应用防火墙等功能。在全球发现13049个此类设备，其中北京市存在500个此类设备。

## 威胁程度

CVE-2020-5902是与2020年7月1日爆出的关于BIG-IP ADC的高危漏洞。通过此漏洞，未授权的远程攻击者通过向漏洞页面发送特制的请求包，可以造成任意 Java 代码执行，进而控制 F5 BIG-IP 的全部功能。威胁包括但不限于: 执行任意系统命令、开启/禁用服务、创建/删除服务器端文件等。该漏洞影响控制面板受影响，不影响数据面板。该漏洞的漏洞等级为**极高**，且影响面为极广泛。

# 现状概述

A screen shot of a smart phone

Description automatically generated

图 1 该漏洞全球影响情况

通过查询网络测绘系统发现，目前存在大规模的BIG-IP设备暴露在公网下。黑客将轻松获取这些网站的大量敏感信息。

其中在北京市发现550个此类设备，详细IP信息见在本文档结尾附件。

# 威胁分析

CVE-2020-5902是BIG-IP流量管理用户界面（TMUI）（也称为配置实用程序）中的重要漏洞。该漏洞的CVSSv3评分为10.0，这是最高的评分。通过BIG-IP管理端口或自身IP公开对TMUI的网络访问时，可以利用此漏洞。成功利用此漏洞将使攻击者获得各种特权，包括执行任意系统命令或Java代码，创建或删除文件以及禁用易受攻击主机上的服务的能力。该通报指出，该漏洞还可能导致整个系统受损。

## 

BIG-IP系统的镜像­，登陆后会要求强制更改密码

图 2 BIG-IP 启动界面

等待系统正常开启后，进入系统IP。之可其Web管理页面（如图3所示）

图 3 网页管理登陆

欢迎如图4所示

图 4 网络管理欢迎界面

3 利用漏洞开启bash。

在浏览器中输入

图 5 开启bash

4

利用如下命令可读取系统passwd文件：

图 6 读取passwd文件

系统passwd文件为系统中的关键敏感文件，存储着系统中的所有用户信息。

5 。

通过如下命令可列出www目录下的所有文件。

图 7 读取www目录下的所有文件

## 漏洞原理解析

研究人员发布的漏洞验证PoC程序利用了fileRead.jsp和tmshCmd.jsp文件发起攻击，但是它们并不是漏洞的起因。此漏洞的本质是利用Apache和后台Java(tomcat)对URL的解析方式不同来绕过登陆限制, 在未授权的情况下，访问后台JSP模块。

F5 BIG-IP的后台服务器对收到了URL请求进行了两次的解析，第一次是httpd(Apache), 第二次是后一层的Java(tomcat)。

最关键的点：在URL在第一次被Apache解析时，Apache关注的是URL的前半段。

https://<IP>/tmui/login.jsp/..;/tmui/locallb/workspace/fileRead.jsp?fileName=/etc/passwd

当Apache在看见前半段是合法URL且是允许被访问的页面时，将其交给了后面的第二层。Apache在这里完全忽视了URL中关键的/..;/。

在URL在第二次被解析时，后面的Java(tomcat)会将/..;/理解为向上返回一层路径。此时, /login.jsp/ 和 /..;/会抵消掉。Tomcat看到的真正请求从

https://<IP>/tmui/login.jsp/..;/tmui/locallb/workspace/fileRead.jsp?fileName=/etc/passwd

变成:

https://<IP>/tmui/tmui/locallb/workspace/fileRead.jsp?fileName=/etc/passwd

另外fileRead.jsp并没有对收到的请求进行身份验证，后台因此直接执行fileRead.jsp, 读取并返回了/etc/passwd文件的内容。

根据以上的思路，可以找出其他利用漏洞的URL, 比如:

https://<IP>/tmui/tmui/login/legal.html/..;/..;/locallb/workspace/fileRead.jsp?fileName=/etc/passwd

其中“https:// <IP>/tmui/tmui/login/legal.html” 和之前的“login.jsp”一样，是一个不需要登陆就能访问的页面。但是因为要向上返回两次，需要用两个/..;/来抵消掉 ”/login/legal.html”。

## 

## 

## 

## 

## 

## 防御措施

在Web服务器的配置文件中添加以下规则：

include ‘<LocationMatch ".\*\.\.;.\*">Redirect 404 /</LocationMatch>’

此规则的含义为：当http服务器在监测到URL中包含..;(句号句号分号)的时候，直接返回404。通过此种方式，可阻止利用漏洞的请求到达后台。

# 总结

此漏洞等级为高，利用难度低，威胁程度高，影响面高。建议使用用户及时安装最新补丁，以免遭受黑客攻击。

# 附件：北京市拥有相似漏洞的网站IP列表

