# CVE-2024-38077的漏洞初步分析以及脚本

## 漏洞背景：

### 漏洞时间线：

2024年05月01日　首次向 Microsoft 报告此安全问题
2024年07月01日　通知 Microsoft 该问题可被利用
2024年07月09日　Microsoft 将其作为 CVE-2024-38077 修复，并标记为利用可能性较低

（微软在7月份的更新中已经修复该漏洞，未自动更新的windows系统均存在该漏洞）

2024年08月02日　将本文草稿发送给 Microsoft 审核
2024年08月09日　未收到 Microsoft 对本文的任何回应
2024年08月09日　文章正式发布

### 漏洞概述：

该漏洞存在于Windows远程桌面许可管理服务（RDL）中，成功利用该漏洞的攻击者可以实现远程代码执行，获取目标系统的控制权。**该漏洞影响所有启用 RDL 服务的 Windows Server服务器，特别是未及时更新 2024 年 7 月微软最新安全补丁的系统**

### windows桌面授权服务（RDL）：

远程桌面许可服务是 Windows Server 的一个组件，用于管理和颁发远程桌面服务的许可证，确保对远程应用程序和桌面的安全且合规的访问。

RDL 服务广泛部署在启用了远程桌面服务的机器上。默认情况下，远程桌面服务仅允许同时使用两个会话。要启用多个同时会话，您需要购买许可证。RDL 服务负责管理这些许可证。RDL 被广泛安装的另一个原因是，在 Windows 服务器上安装远程桌面服务 (3389) 时，管理员通常会勾选安装 RDL 的选项。这导致许多启用了 3389 的服务器也启用了 RDL 服务。

在审计RDL服务之前，我们进行了网络扫描，以确定RDL服务在互联网上的部署情况。我们发现至少有17万个活跃的RDL服务直接暴露在公共互联网上，而内部网络中的数量无疑要大得多。此外，RDL服务通常部署在关键业务系统和远程桌面集群中，因此RDL服务中的预认证RCE漏洞对网络世界构成了重大威胁

## 组件安装：


## 组件服务：


## 影响版本：

```
Windows Server 2025 Preview

Windows Server 2012 R2 (Server Core installation)

Windows Server 2012 R2

Windows Server 2012 (Server Core installation)

Windows Server 2012

Windows Server 2008 R2 for x64-based Systems Service Pack 1 (Server Core
installation)

Windows Server 2008 R2 for x64-based Systems Service Pack 1 (Server Core
installation)

Windows Server 2008 R2 for x64-based Systems Service Pack 1

Windows Server 2008 R2 for x64-based Systems Service Pack 1
```

```
Windows Server 2008 for x64-based Systems Service Pack 2 (Server Core
installation)

Windows Server 2008 for x64-based Systems Service Pack 2 (Server Core
installation)

Windows Server 2008 for x64-based Systems Service Pack 2

Windows Server 2008 for x64-based Systems Service Pack 2

Windows Server 2008 for 32-bit Systems Service Pack 2 (Server Core installation)

Windows Server 2008 for 32-bit Systems Service Pack 2 (Server Core installation)

Windows Server 2008 for 32-bit Systems Service Pack 2

Windows Server 2008 for 32-bit Systems Service Pack 2

Windows Server 2016 (Server Core installation)

Windows Server 2016

Windows Server 2022, 23H2 Edition (Server Core installation)

Windows Server 2022 (Server Core installation)

Windows Server 2022

Windows Server 2019 (Server Core installation)

Windows Server 2019
```

# 漏洞分析：

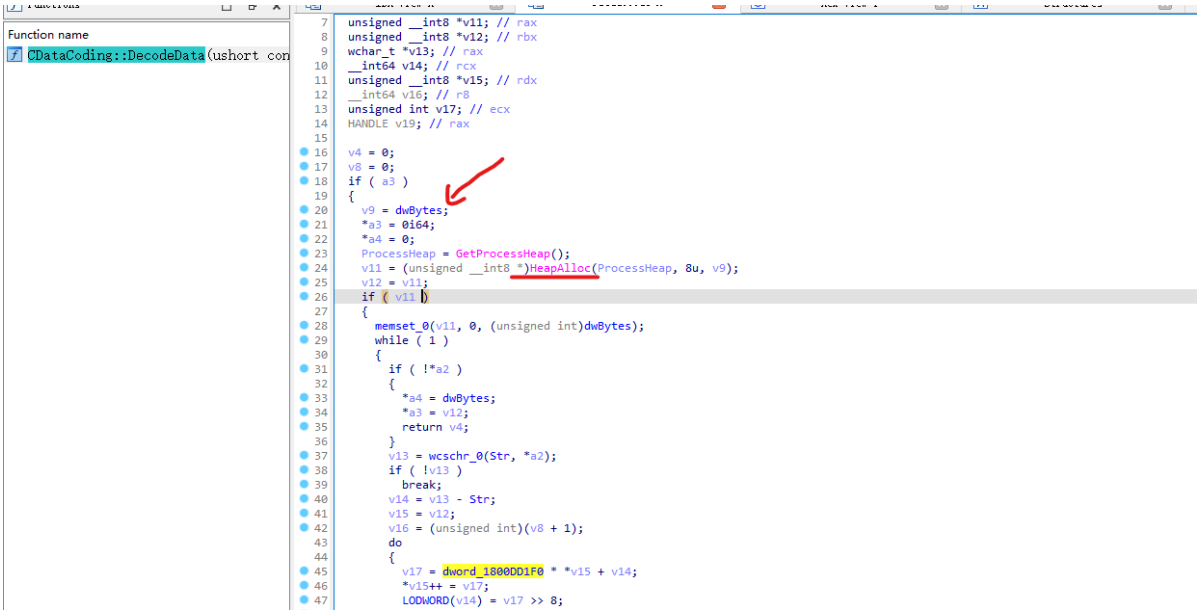
根据漏洞发布者的描述：

1、该程序中 `CDataCoding::DecodeData`，分配一个固定大小的缓冲区（21字节），然后计算并填充用户控制的长度缓冲区，从而引起堆溢出。因此，想要在目标主机上执行命令，需要先泄露dll基地址，然后构造后门才可进行命令执行。

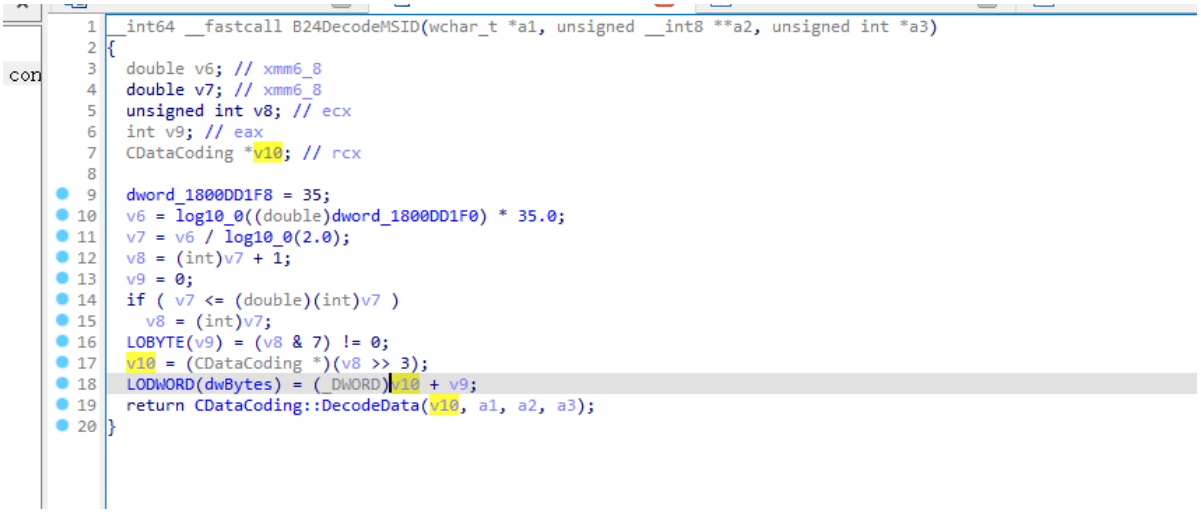
2、此漏洞是Windows Server RDL服务中的一个堆溢出漏洞（CWE-122），由于在解码用户输入的许可密钥包时，未正确检验解码后数据长度与缓冲区大小之间的关系，导致堆溢出。未经授权的远程攻击者通过向存在漏洞的Windows Server发送特制的数据包来利用此漏洞，成功利用此漏洞可在该目标服务器上执行任意代码。

**分析过程：**

我们打开ida找到作者所给出的函数：




这里确实有对堆的分配行为，从进程堆中分配了内存，大小由v9进行决定，而v9由dwBytes决定。跟进dwBytes，走到如下函数：



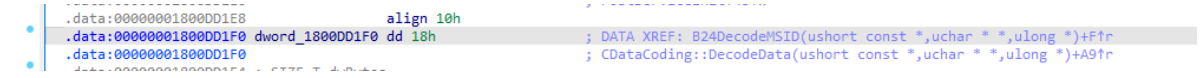

在上述代码中：

- `v10` 是通过右移 `v8` 计算得到的 `CDataCoding` 指针值。
- `v9` 是一个标志位或调整值，决定了最终 `dwBytes` 的大小。

因此，`dwBytes` 的值在 `B24DecodeMSID` 函数中被动态计算和设置，具体值取决于 `v8` 和 `v9` 的计算结果。

在这里面：dword_1800DD1F8 = 35

跟进dword_1800DD1F0：




定义了初始值为24

因此，计算出 在 `dword_1800DD1F0 = 24` 的情况下，`dwBytes` 的最终值为 **21**。因此这里的堆地址大小是固定的，大小为21,即为作者所说;

回到原函数，对堆分配的内存是用来存储解码后的数据的,而所解码的内容即为用户的可控部分,因此出现了堆的溢出问题.

```
34          *a3 = v12;
35          return v4;
36        }
37        v13 = wcschr_0(Str, *a2);
38        if ( !v13 )
39          break;
40        v14 = v13 - Str;
41        v15 = v12;
42        v16 = (unsigned int)(v8 + 1);
43        do
44        {
45          v17 = dword_1800DD1F0 * *v15 + v14;
46          *v15++ = v17;
47          LODWORD(v14) = v17 >> 8;
48          --v16;
49        }
50        while ( v16 );
51        if ( (_DWORD)v14 )
52          v12[++v8] = v14;
53        ++a2;
54      }
55      v4 = 13;
56      v19 = GetProcessHeap();
57      HeapFree(v19, 0, v12);
58    }
59    else
```

因此,对于该漏洞,是由于对于用户的输入没有进行任何的限制导致的漏洞.很经典的溢出问题.

在原作者的poc中,没有直接在3389来进行利用,而是通过其他服务来发送恶意数据流,比如135端口上的服务.

# 漏洞验证:

## 验证思路:

综上分析,当输入的数据过长的时候造成溢出,因此没有直接的办法对漏洞进行poc验证.但在7月份补丁的更新之前,只要该服务开启,任意用户都可以远程过程调用（RPC）,因此我们可以直接发送RPC请求来获取该服务器版本.而刚好,微软在7月份的更新中,限制了这一行为,因此可以通过这个方式来进行漏洞的验证.

```
rpc_conn = TLSRpcConnect()
res_rpc_conn = dce.request(rpc_conn)
ctx_handle = res_rpc_conn["ctx_handle"]
get_version = TLSRpcGetVersion()
get_version["ctx_handle"] = ctx_handle
get_version["version"] = 3
res_get_version = dce.request(get_version)
version = res_get_version["version"]
print("[+] Get Server version: 0x{:x}".format(version))
```

验证poc如下:

```
-----peppa
[+] Get Server version: 0x60000a04
```

只要能返回版本号即可确定存在该漏洞;更新补丁之后,显示了调用rpc没有权限:

```
Traceback (most recent call last):
  File "D:\网络安全\集成工具\mytool\杂乱无章的工具箱\经典漏洞利用工具\CVE-2024-38077-POC-远程利用工具\CVE-2024-38077-EXP.py",
    connect_to_license_server("192.168.6.132")
  File "D:\网络安全\集成工具\mytool\杂乱无章的工具箱\经典漏洞利用工具\CVE-2024-38077-POC-远程利用工具\CVE-2024-38077-EXP.py",
    res_rpc_conn = dce.request(rpc_conn)
  File "C:\Users\peppa\AppData\Roaming\Python\Python310\site-packages\impacket\dcerpc\v5\rpcrt.py", line 859, in request
    answer = self.recv()
  File "C:\Users\peppa\AppData\Roaming\Python\Python310\site-packages\impacket\dcerpc\v5\rpcrt.py", line 1323, in recv
    raise DCERPCException(rpc_status_codes[status_code])
impacket.dcerpc.v5.rpcrt.DCERPCException: rpc_s_access_denied
PS D:\网络安全\集成工具\mytool\杂乱无章的工具箱\经典漏洞利用工具\CVE-2024-38077-POC-远程利用工具>
```

# 漏洞修复:

1.目前微软已经发布补丁,及时更新;

2.对于无法线上更新的主机,可下载补丁进行修复,

3.无法更新的主机可以禁用rdl该服务.