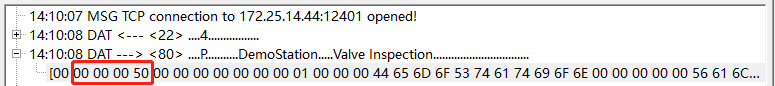
# 程序信息

|  |  |
| --- | --- |
| 名称 | IGSS Data Server |
| 文件名 | IGSSdataServer.exe |
| 端口/协议 | 12401/TCP |
| 探测包 | [16 00 01 00 34 12 0D 00 00 00 00 00 00 00 01 00 00 00 01 00 00 00] |

**探测依据：**

查看响应包的第2位至第5位是否为\x00\x00\x00\x50，如下图所示：



目前已经通过验证的版本：

V9，V10，V11，V12，V13，V14

# 程序信息

|  |  |
| --- | --- |
| 名称 | IGSS Data Collection Module |
| 文件名 | dc.exe |
| 端口/协议 | 12397/TCP，12399/TCP |
| 探测包 | 12397端口：[1C 00 00 00 00 00 00 00 00 70 41 41 65 00 00 00 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 00 00]  12399端口：[0C 00 00 00 00 40 00 00 61 61 61 61] |

**探测依据：**

12397端口：

判断返回的数据包首部两字节是否为\x20\x00，如下图所示：



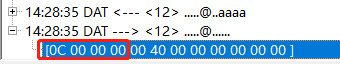
目前已经通过验证的版本：

V9，V10，V11，V12，V13，V14

12399端口：

判断返回的首部四个字节是否数据包总长的小端序表示，也可判断是否为

\x0C\x00开头，因为目前来看都是返回12个字节的数据包：



目前已经通过验证的版本：

V9，V10，V11，V12，V13，V14

# 程序信息

|  |  |
| --- | --- |
| 名称 | IGSS32 ODBC Server |
| 文件名 | Odbcixse.exe|Odbcixv9se.exe|Odbcixv10se.exe |
| 端口/协议 | 20222/TCP |
| 探测包 | [00 00 00 00] |

**探测依据：**

查看响应包是否为\x00\x00\x00\x02\x00\x03，如下图所示：



目前已经通过验证的版本：

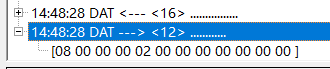
V9，V10，V11，V12，V13，V14

# 程序信息

|  |  |
| --- | --- |
| 名称 | IGSSupdater.exe |
| 文件名 | IGSSupdateservice.exe |
| 端口/协议 | 12412/TCP（V12），12413/TCP（V13），12414/TCP（V14） |
| 探测包 | [0E 00 00 00 01 00 00 00 08 00 00 00 00 00 00 00] |

**探测依据：**

查看响应包是否以\x08\x00\x00\x00开头，如下图所示：



目前已经通过验证的版本：

V12，V13，V14

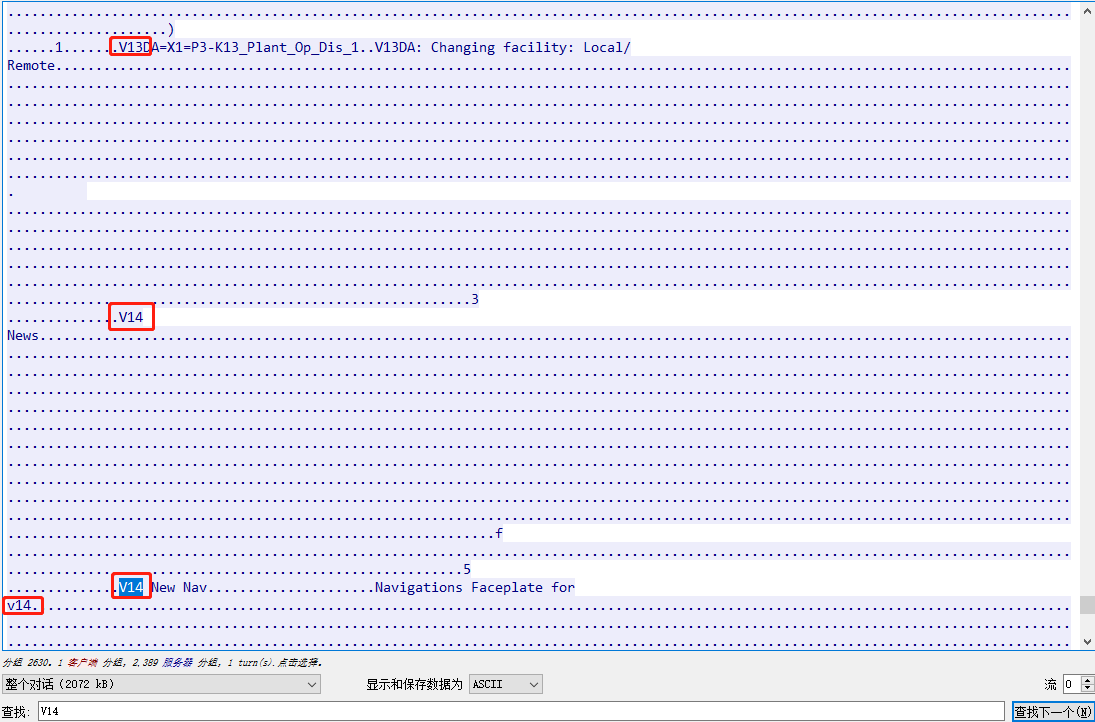
V9，V10，V11版本未发现此进程

# 版本区分：

研究发现，探测包：

**\x12\x00\x01\x00\x34\x12\x03\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00**

到目标IP的IGSSDataServer监听端口，也就是12401端口，会返回许多个较大的数据包，其数据包中包含众多类似于”Vxx”似乎是版本信息的字符串，如下图所示：



在一个特定的版本中，该探测包的响应数据可能包含V9、V10、V13等众多跟版本不符的数字。研究发现，响应数据中的最后一个”Vxx”字符串是与目标IGSS版本号吻合的，那么我们尾部开始搜索第一个”Vxx”字符串，提取出来的数字就为目标IGSS的版本号，写成Python3代码如下所示：

soc = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
payload = **b'\x12\x00\x01\x00\x34\x12\x03\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00\x00'  
try**:  
 con = soc.connect((target\_ip, dataserver\_port))  
 soc.send(payload)  
 result = **b''  
 while True**:  
 data = soc.recv(10240)  
 **if** len(data) == 0:  
 **break** result += data  
 result = result[::-1]  
 pat = re.compile(**b'(\d{1,2})V'**)  
 ver = pat.search(result)  
 print(**"目标IGSS版本为："**, ver.group(1)[::-1].decode(**'ascii'**))  
**except** Exception **as** e:  
 print(e)

目前已经通过验证的版本：

V9，V10，V11，V12，V13，V14