|  |  |  |  |
| --- | --- | --- | --- |
| V1.0 | 2018年7月10日 | 开始试用 | 刘强 |
| V1.1 | 2018年7月12日 | 增加UiPath使用经验 | 刘强 |
| V1.2 | 2018年7月15日 | 增加UiPath使用经验 | 刘强 |
| V2.0 | 2018年7月19日 | 修改RAT工作目录，添加output转存日期，工程模块化 | 刘强 |

目录

[1. 测试环境准备 3](#_Toc520987912)

[2. 添加一个新的RAT（配置RAT） 4](#_Toc520987913)

[3. 为新的RAT录制PHF 5](#_Toc520987914)

[3.1. 任务要求 5](#_Toc520987915)

[3.2. 相关经验 5](#_Toc520987916)

[3.3. 高级功能 5](#_Toc520987917)

[3.3.1. 窗口定位 5](#_Toc520987918)

[3.3.2. 列表定位 6](#_Toc520987919)

[3.3.3. 经典主题 8](#_Toc520987920)

[3.3.4. 不可见控件（树形控件和文本控件） 8](#_Toc520987921)

[3.3.5. Ui Explorer无法定位的控件 8](#_Toc520987922)

[3.4. 一些问题 9](#_Toc520987923)

# 测试环境准备

* 1. Windows 7，关闭UAC，防火墙，360，更新powershell执行策略。进入powershell输入：set-executionpolicy remotesigned。输入法换成英文。切换到经典桌面主题，即不要任何附加效果。
  2. 安装WinRAR，更新` tools\apps\update\_address\_map.bat`的代码，需要更新的有：WinRAR的地址。
  3. 更新Collector软件，<http://118.126.94.181/APT/apt_collector_release>，永远使用最新版本的软件。
  4. 运行` tools\apps\update\_address\_map.bat`更新Collector的address map（保证网络连接，1~2分钟）。成功如图 1所示。每次测试前只需要更新一次。

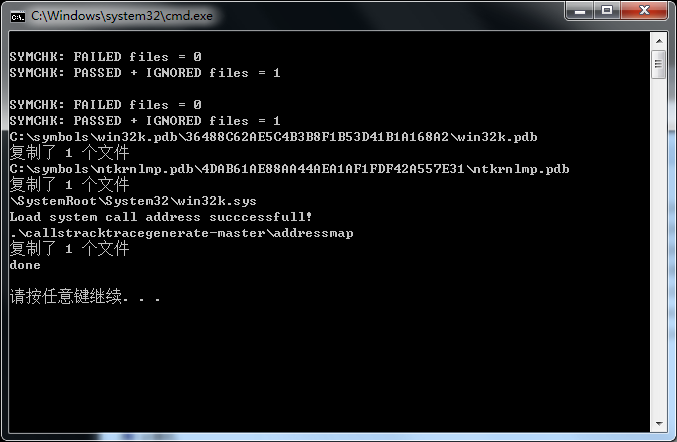


图 1‑1

* 1. 安装Python2.7。
  2. 安装UiPath Studio（10.214.148.12:/home/liuqiang/UiPathStudioSetup.exe，需要连网，按提示输入邮箱即可，如cyruscyliu@gmail.com）。打开UiPath Studio找到包管理器，如图 2所示。打开包管理器，在Available中搜索UiPath.Script.Activities，点击install安装。

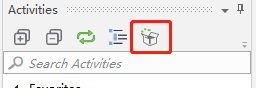


图 1‑2

# 添加一个新的RAT（配置RAT）

* 1. 按下表所示路径和文件名放置RAT

|  |
| --- |
| c:\tools\rats\rat\_name\_v[[1]](#footnote-1)\server.exe（需要提前生成，不要开安装） |
| c:\tools\rats\rat\_name\_v\rat\_name.exe |

* 1. 配置UiPath工程生成器
     1. 配置configuration.py

|  |  |
| --- | --- |
| ROOT：“c:\\tools” | 工具包路径 |
| COLLECTOR\_NAME：'ETWDataCollectorRedesigned.exe' | Collector名（按需[[2]](#footnote-2)） |
| LINE\_NUMBER：12 | （按需[[3]](#footnote-3)） |

* + 1. 使用UiPath Studio录制除了rat启动和终止以外其他需要记录的部分，将这些集成在一个Main.xaml文件中，如果想要单独记录每一个行为可以将不同的功能存放到各自的XAML文件中，将生成的xaml文件放在c:\tools\uipath\_projects\rat\_name文件夹下。
    2. 自动化执行RAT并收集数据（每添加一个新的RAT）

|  |
| --- |
| c:\python2.7\python.exe uipath\_project\_render.py c:\tools\rat\_name\_v\rat\_name.exe [-c] [-d] |

示例见tools\render\_sample.bat，其中默认使用collector收集数据，使用-c选项可以选择不使用collector仅启动RAT，使用-d选项分别存储Keylogger,、RemoteDesktop、 RemoteShell、RemoteAudio、Download\_Execution等行为的数据，前提是使用UiPath Studio录制的时候也需要单独录制并存储成这些文件。

程序结束之后，如果选择了使用collector收集数据那么数据会被存储在output文件夹下，以rat版本号和时间戳命名，如图2-1所示：

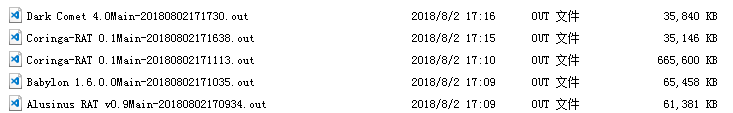


图 2‑1

具体运行情况如图2-2所示：

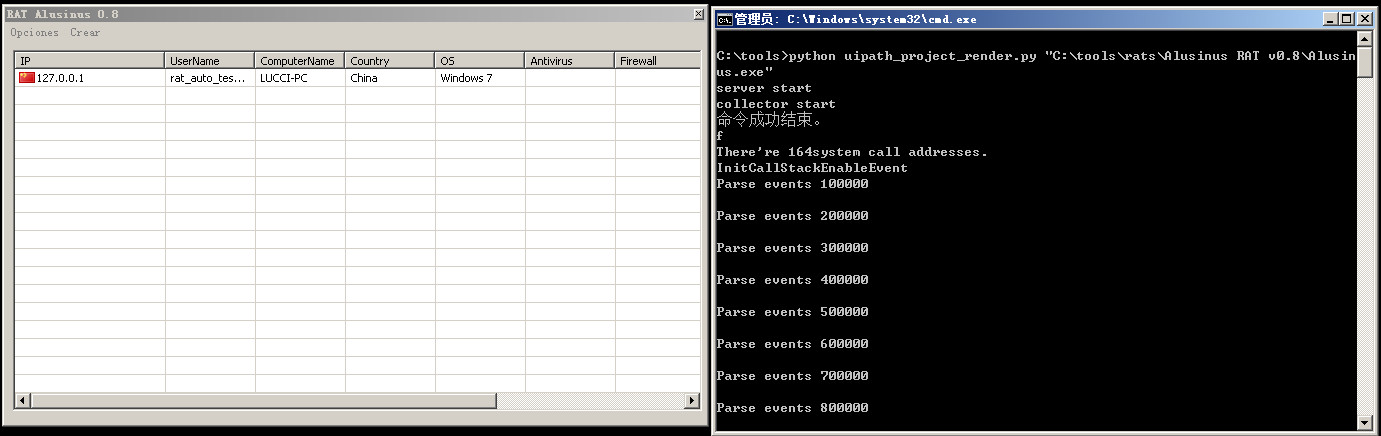


图 2-2

* 1. 批量自动化测试

使用processing.py脚本完成批量测试，需要提前将需要测试的RAT文件路径写在rat\_list.txt文件下如图2-3所示，然后执行如下命令便可完成列表中所有RAT的自动化测试与数据采集工作。

|  |
| --- |
| c:\python2.7\python.exe processing.py |



图 2-3

具体运行情况如图2-4所示：

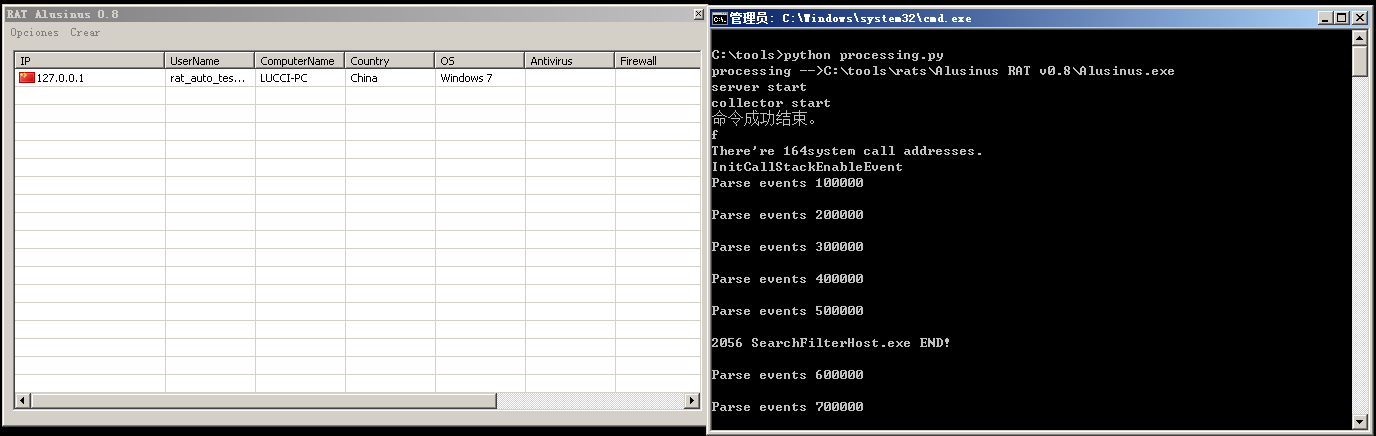


图 2-4

# 为新的RAT录制PHF

## 任务要求

* + 1. 录制5个PHF，分别是Remote Desktop、Remote Audio、Remote Shell、Download & Execution and Keylogger。
    2. 每录制一个PHF都需要保证该PHF执行成功，并转存trace文件。对待操作不流畅、功能不完善的RAT，应记录在《RAT路径及命名一览表》中，并作出相应评级。
    3. output文件已自动转存。

## 相关经验

* + 1. 当RAT自动弹网页、资源管理器的时候，请使用Set Focus获取RAT的焦点。
    2. “点击鼠标右键”录制成“点击”鼠标左键，而后利用UiPath Studio调整改动作的属性为右键。
    3. 利用F2暂停录制，点击鼠标右键弹出相关菜单。
    4. 录制按钮时，请先将鼠标放置到按钮上，保证录制过程中，程序能正常执行。比如，录制“关闭窗口”时，按F2，将鼠标移至右上角，关闭按钮会发生颜色变化，录制时，点击鼠标左键，该动作被录制的同时，该窗口也应该被关闭。
    5. 点击动作录制不成功（执行不成功或者录成了Select Item），需要重新录制该动作。首先，使用录制面板上的“录制点击功能”，如图 5所示。再使用Click Image功能。再使用Click OCR Image功能。一般，录制面板即可解决问题。

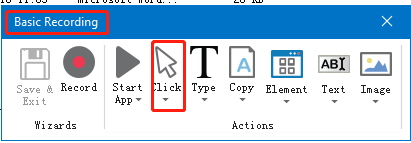


图 3‑1

* + 1. 重复操作录制一次，复制以下即可。粘贴时，选中活动之间的倒三角。
    2. 运行时可利用disable功能就行部分调试
    3. 运行时桌面窗口不要放太多

## 高级功能

UiPath依靠录制时的情况进行重现，通用性较差，Andong Chen优化了UI元素选择的策略，使得自动化程序具有通用性。

### 窗口定位

首先是在窗口定位方面，取消关于title、name等项的设置。实测中发现很多RAT窗口的标题并非为静态的，如Bozok的标题带有当前在线的机器数量，如图 3‑2所示，使用录制或Selector Editor得到的结果为<wnd app='bozok.exe' cls='TForm1' title='Bozok 1.4.3 - 0 online' />，如果在线机器数量发生改变，显然无法定位窗口。但一般GUI编程一个类对应一种窗口，而对RAT测试一般只对同一功能打开一个窗口，因此仅通过cls字段就能够准确定位窗口。

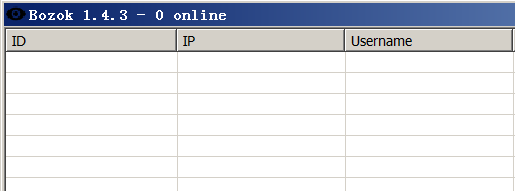


图 3‑2

在录制或者使用UI Explorer记录操作时，应检查生成的xml，对于可能的非静态内容（包括但不限于时间、日期、主机名、IP地址、在线数量等），做注释处理。

通过上述方法可以完成对静态UI组件的自动化操作。

### 列表定位

RAT中大量使用list控件，内容动态更新，主要用于列出在线主机或者日志。Ui Explorer提供的列表项的定位方法是通过列表项名字或index进行定位，显然不适用于更广泛的需求。例如DarkComet，可以看到UiPath是使用了list item的第一项值作为name进行定位的，如图 3‑3所示。

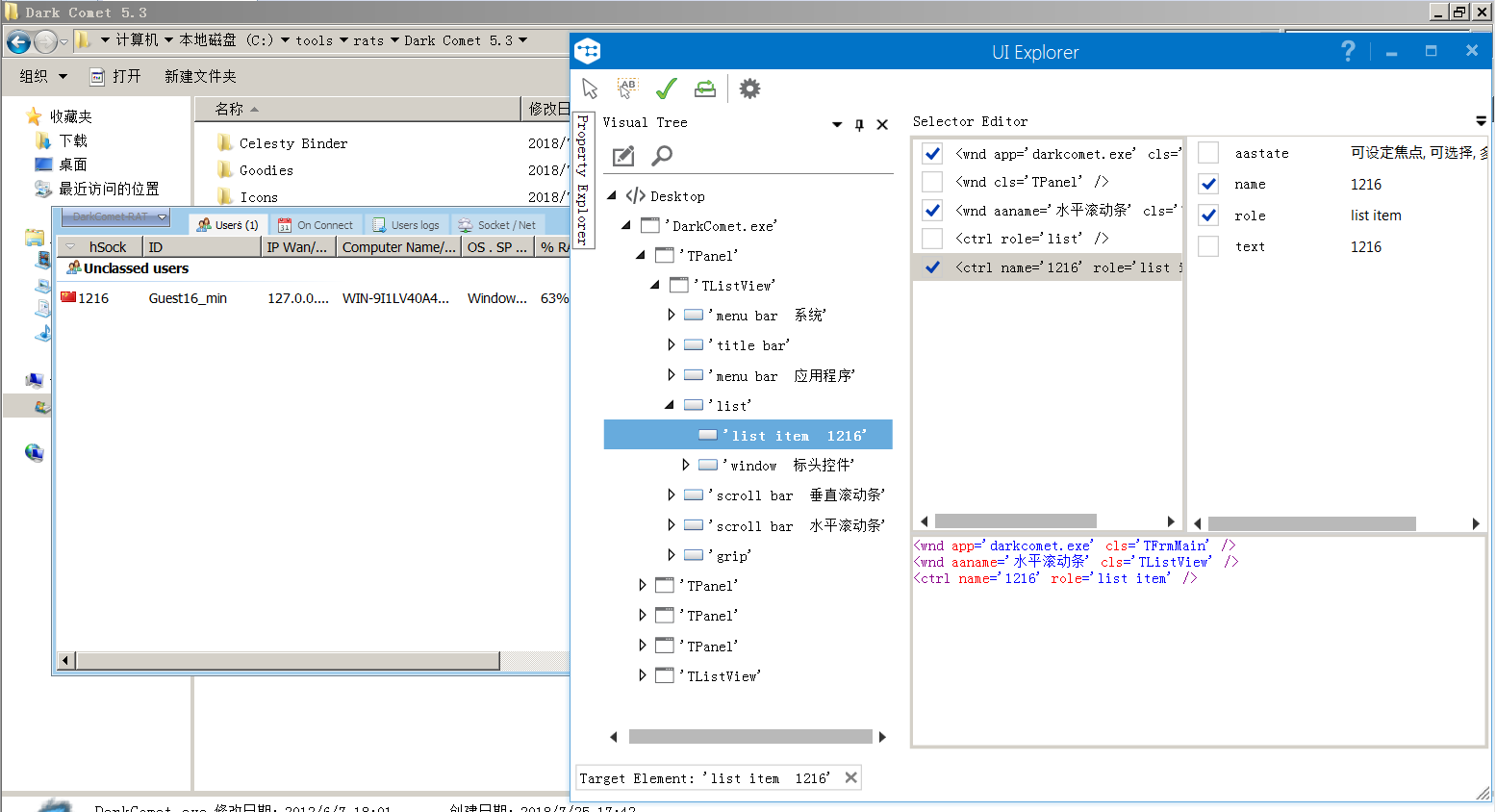


图 3‑3

但这一值不可能完全固定，即使是同一主机，在DarkComet中的hSock值仍会改变，如图 3‑4所示，因而利用name定位list item的做法不具有普适性。

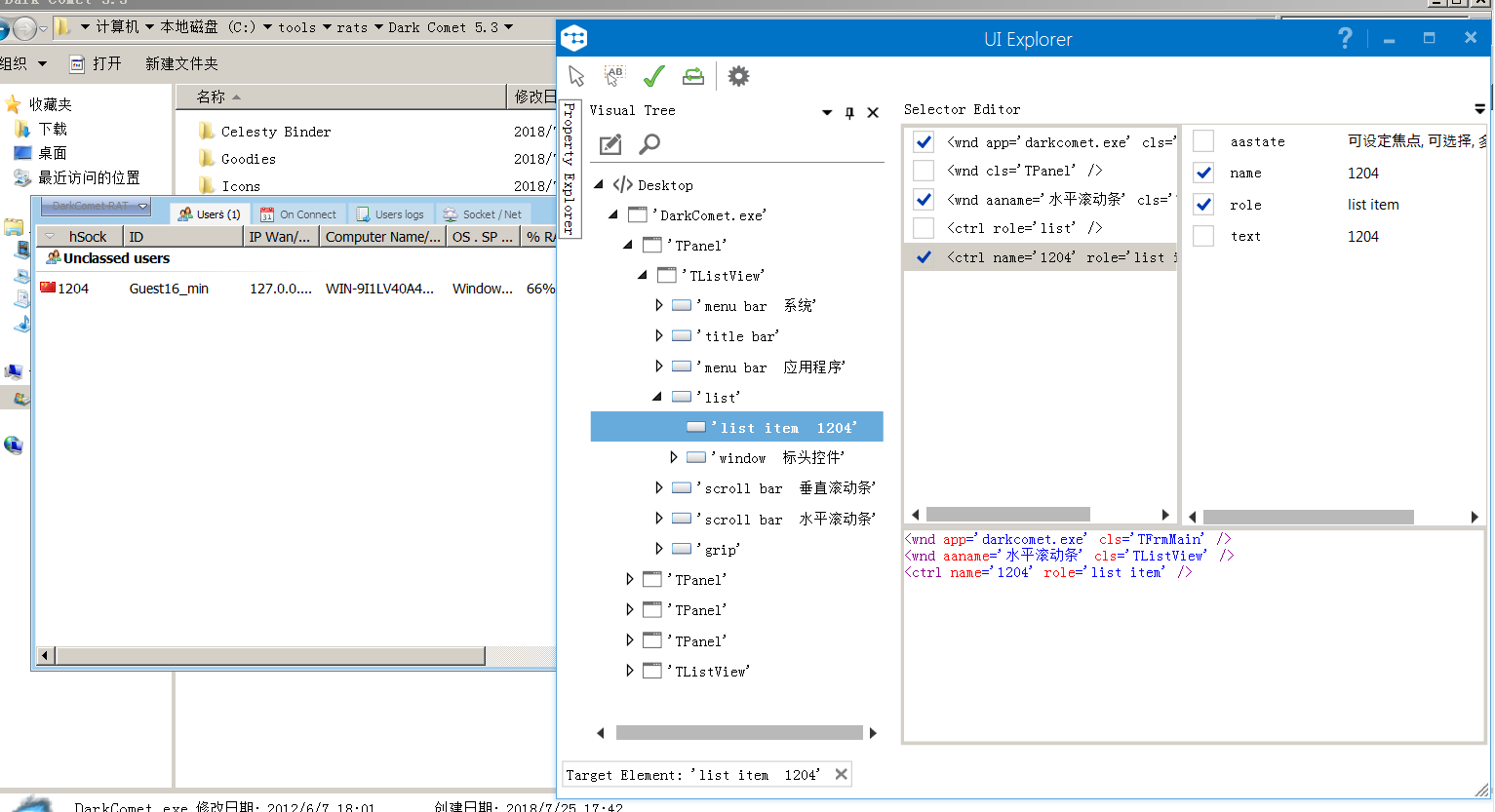


图 3‑4

但从Ui Explorer可以看到list item是ListView的子元素，考虑使用UiPath提供的Find Children Activity，直接定位ListView，得到所有子元素，并利用其filter属性筛选列表项

<ctrl role=’list item’ />。定义IEnumerable<UiElement>变量hostElements保存查找到的列表项，即每个主机项Element。

利用Foreach Activity遍历hostElements，可得到每一个主机项对应的列表项控件，如图 3‑5所示。在后续的过程中，对该主机控件的操作（例如点击，设置焦点等）只需定义target Element 为hostElement而不需要通过Selector再去进行选择，如图 3‑6所示。这种做法相较利用键盘控制选择列表项的好处在于，可以直接取得所有列表项而不需要去额外判断列表元素的数量。且因为已经拿到列表项控件的引用，所以即使在失焦时也能对控件进行操作，具有更强的容错性。



图 3‑5

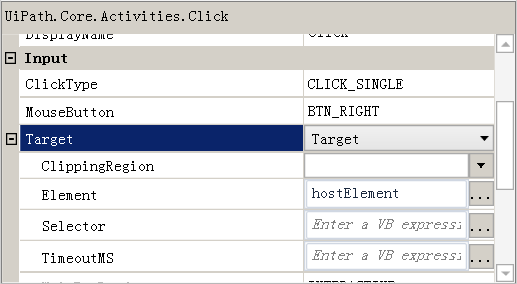


图 3‑6

此外，除了用于主机列表项的定位，在树形目录，或者其他难定位的控件，均可利用Find Children定位一个范围更大且易定位的控件继而获得其内部组件的方式进行元素的动态定位。

### 经典主题

此外由于Win7默认主题的效果的影响，UiPath并不总能有效定位组件，可以通过使用经典主题的方式解决。

### 不可见控件（树形控件和文本控件）

对于可见控件，只需定位到控件执行点击等操作。但对于不可见控件，如未展开的树形控件、在列表当前页面下方的项目等，需要进行模拟键鼠操作，将树形控件展开或拖动滚动条，由于已经基本可以获得所有控件的Element，可以结合VB.net编写程序，特殊情况特殊处理。

目前遇到的树形结构主要是日志文件，如Cloud Net但基本上日志是按日期排序，所以最下或最上方的list item一般为需要的日志项，可以搭配home/end/RIGHT键将树展开，使控件可见。但对于非排序的结构，可能需要编写VB.net程序做特殊处理。

对于不可见列表项，可以利用Selector记录滚动条控件，结合whille循环和Element的状态，当Find Children找到的Element处于不可见状态时，利用Page Up/Down，直到Element可见。

### Ui Explorer无法定位的控件

由于Ui Explorer自身的原因，当控件所处的层次较深时，在录制时UiPath可能无法正确定位控件，只能定位到其上层组件，例如Babylon RAT的主机项。但使用Find Element/Find Children等Activity时可以正确获得控件对应的Element并提取Selector。据此，编写一辅助工具，使用时只需定位上层控件，而后，上层控件内部的组件会依次高亮，相应的selector会填入剪贴板。当所需定位的控件高亮后停止，即得到所需的selector。

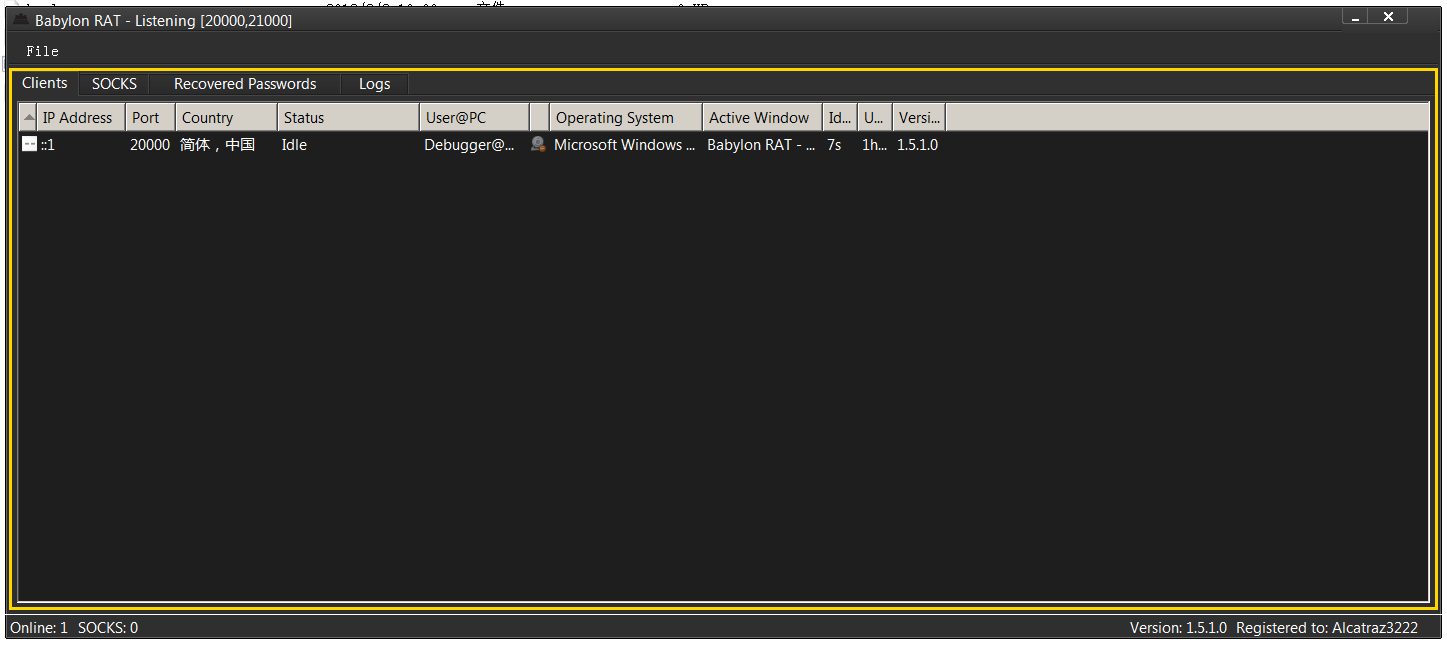


图 3‑7

## 一些问题

* + 1. 弹出Do you want to use an Anchor? 选中Do not show…，点击No。当前动作已经被录制。

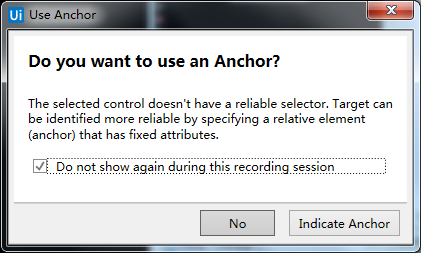


图 3‑8

* + 1. PID更新成功，弹出“找不到数据收集器”并EnableTrace() failed with 5。原因是未使用管理员权限。

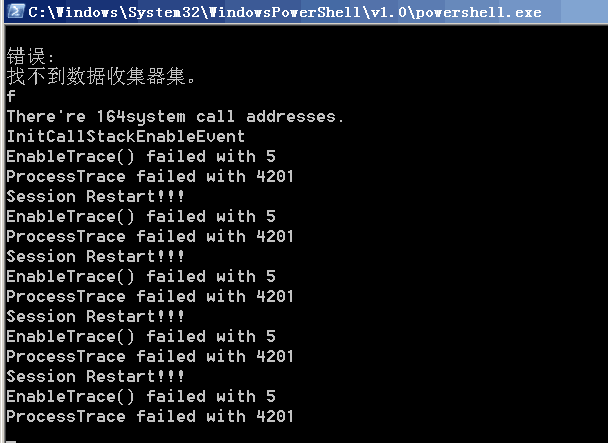


图 3‑9

* + 1. 使用管理员权限时，UiPath可能无法获得程序控件的详细信息，导致控件查找失败。在UAC启动的情况下，无论是从普通用户权限提升至管理员权限，还是从管理员用户切换到普通用户，都需要人工介入，点击允许使用管理员身份或者输入用户密码。为实现无人值守，通过关闭UAC解决。

1. 带版本号的RAT名字 [↑](#footnote-ref-1)
2. Collector更新后主程序的名字有时候会变换，现添加了版本号，请做好区分。 [↑](#footnote-ref-2)
3. Collector更新后，需要确定`output\_whitelist\_processid `的行号，并将行号减1赋值给LINE\_NUMBER。 [↑](#footnote-ref-3)