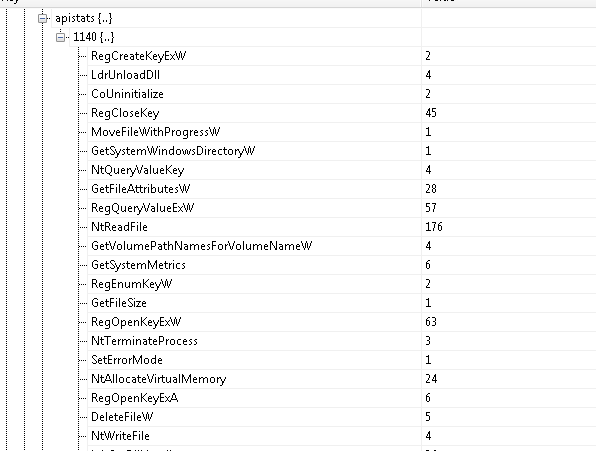
## 背景

通常对病毒使用API的认识都是基于病毒工程师的经验，但是我们并不知道每个API对于是否是病毒的贡献有多大。通过对大量病毒样本的统计分析，我们可以得出相关的数据。

从网上下载3000个样本，1500个病毒，1500个无毒。用沙箱进行扫描，得到样本调用API和是否是病毒。 如下图所示



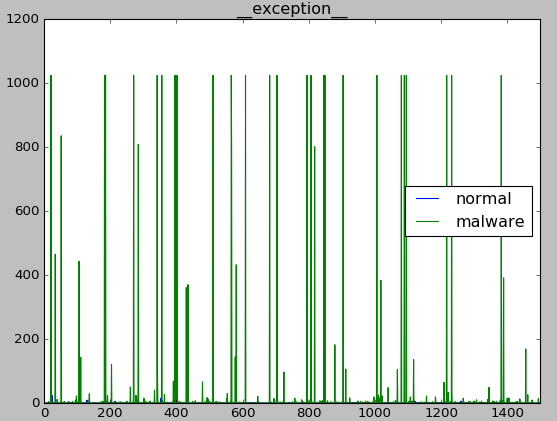
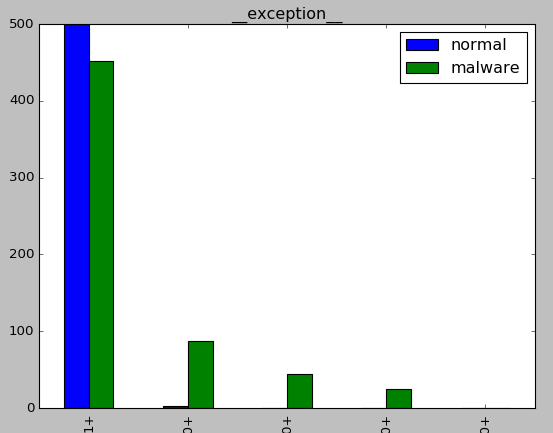
取API的动态调用次数作为每个样本的特征，病毒类标记为1，非病毒类标记为0。如下所示:

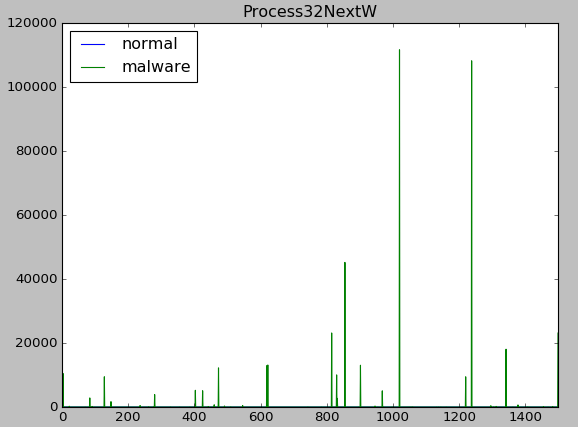
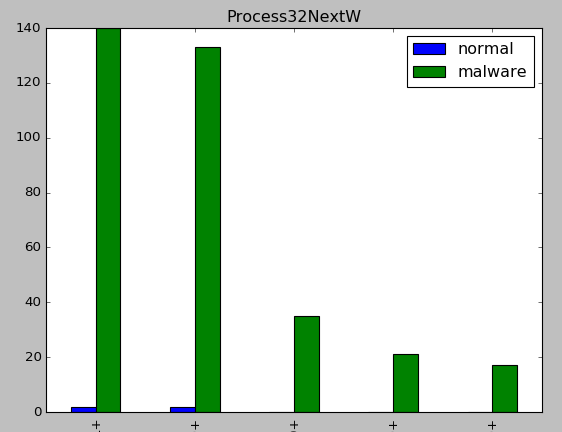
{‘LdrUnloadDll’:4, ‘RegCloseKey’:45,…} 1

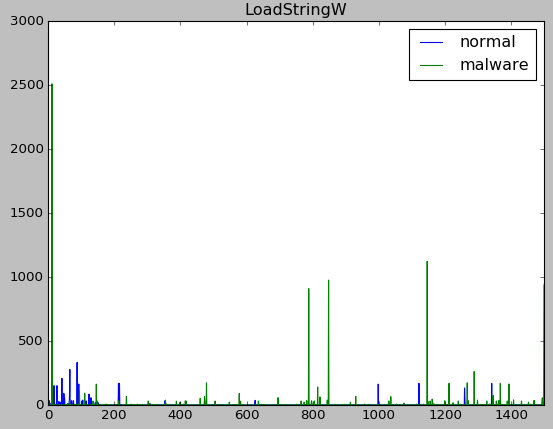
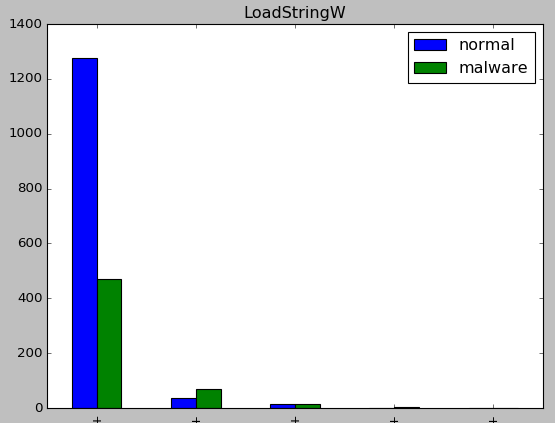
{‘GetFileSize’:1, ‘NtWriteFile’:4, ….} 0

## 分析

将样本的API调用次数表示到折线图和条形图上，蓝色表示正常样本，绿色表示病毒样本。如下图所示：

条形图按API调用的次数分布到5个区间1+（1～10），10+（10～100），100+（100～1000），1000+（1000～5000），5000+（5000～）。

通过对大量样本的图形的观察发现，病毒比较常用的API（如：WriteProcessMemory，Process32NextW等）不管是折线图还是条形图，绿色线条所占的比例都很大。

病毒使用率低的API，蓝色线条占的比例比较大。

有些API的调用，折线图绿线比例大，条形图蓝线比例比较大。这种情况下病毒对API的调用次数会很多（会有很多成千上万次的调用），普通样本对API调用次数很少（一般不超过100次）。

因此我们可以计算折线图和条形图中绿色像素所占的比例来评估API对病毒的影响。

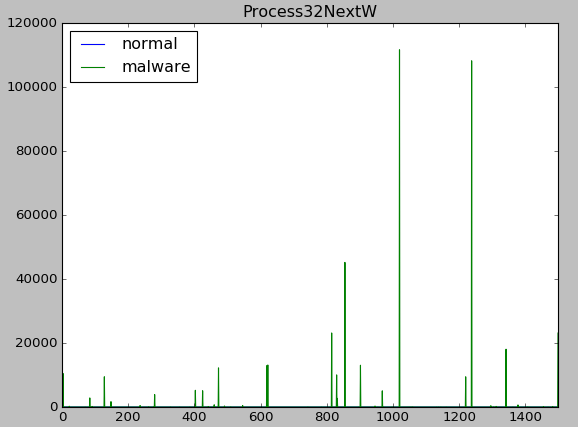
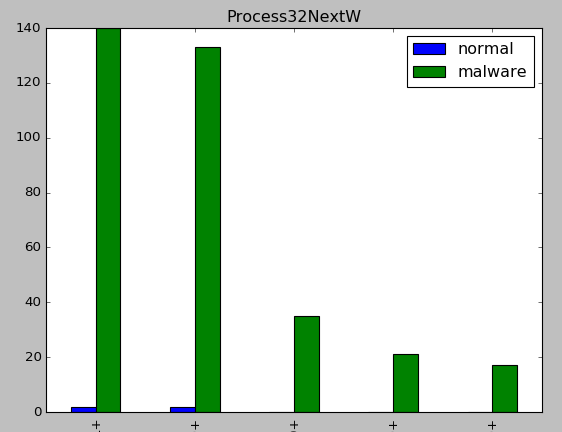
## 方法

P1 = 折线图占用率：绿色像素/(绿色像素+蓝色像素)

P2 = 条形图占用率：绿色像素/(绿色像素+蓝色像素)

1. 计算API对病毒分类的贡献（P1\*P2）

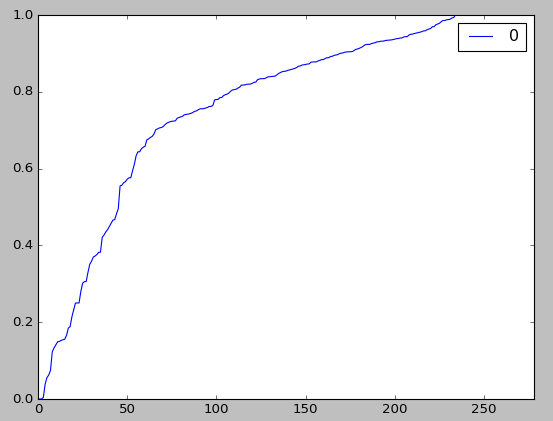
对于最能表示病毒特征的API，其被病毒调用的概率要比被正常样本调用的概率大的多，同时高频调用的次数也比正常样本大的多。如Process32NextW

所以P1，P2的值也会很大。

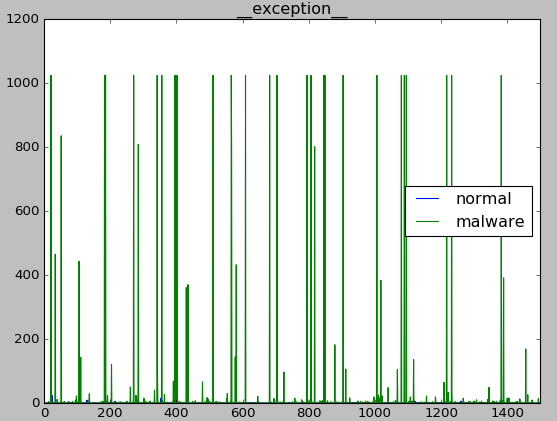
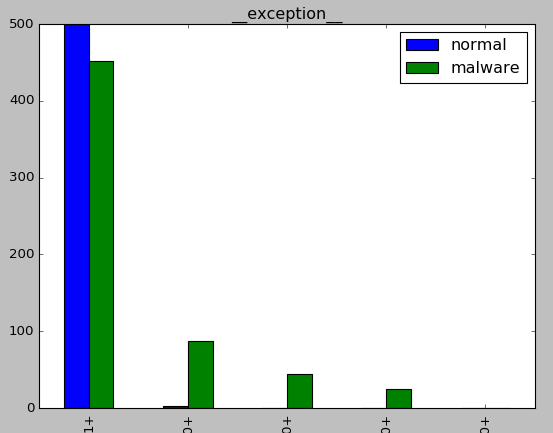
对于最不可能被病毒调用的API，P1，P2的值都会很小。

因此可以用P1\*P2来表示API对病毒分类的贡献，进行排序后，数值会往两端移动。



贡献度排序曲线

1. 计算API的高频调用对病毒分类的贡献(P1-P2)

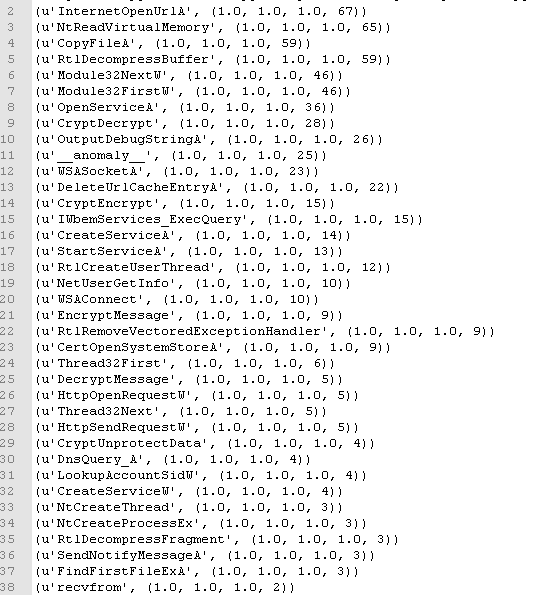
对于\_\_exception\_\_这样的，P1值很大，但是P2值比较小。其高频次调用的情况下是病毒的概率很高，无法单纯的从是否调用来判断是否为病毒。

用P1-P2来评价高频次调用。

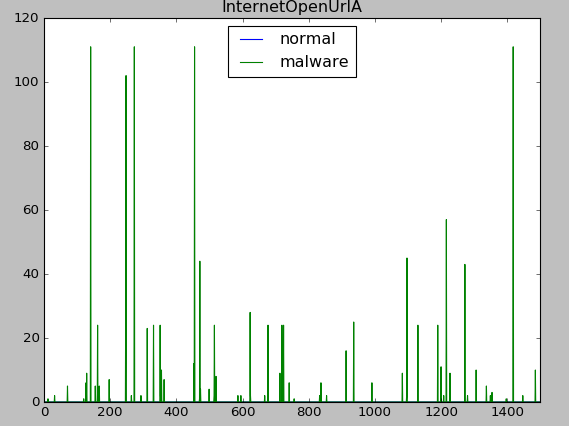
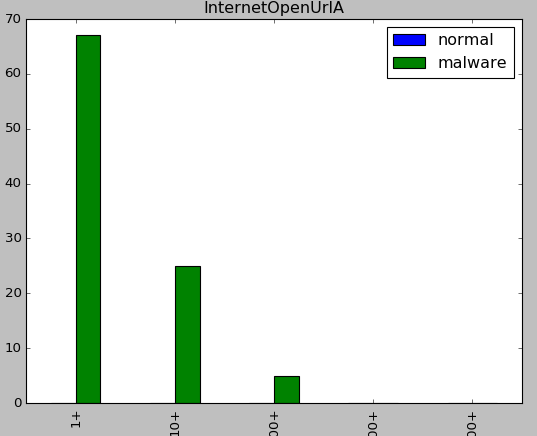
## 结果

1. 计算P1\*P2，进行排序，部分结果如下。

数据结果分别为（API名，（贡献值，API调用次数概率，调用API的样本为病毒的概率，病毒样本数量））



贡献值为1的API，没有被正常样本调用过。（可能是由于样本数量太小造成的，特别是病毒样本数量只有几次的API）。其图像如下：

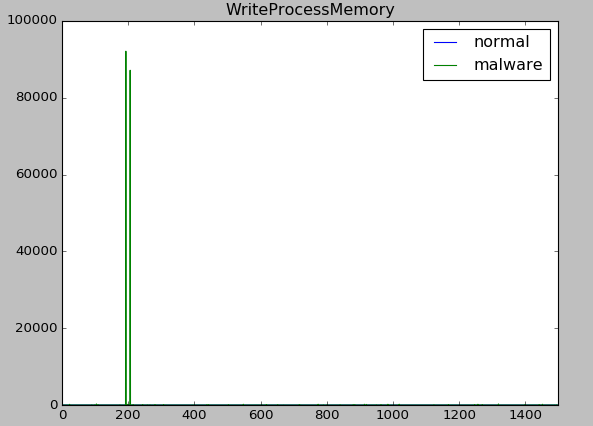
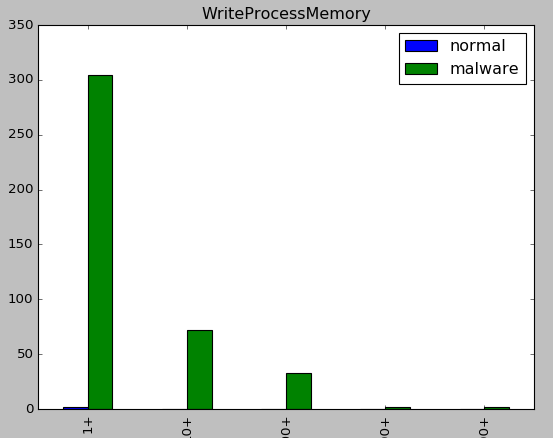
 

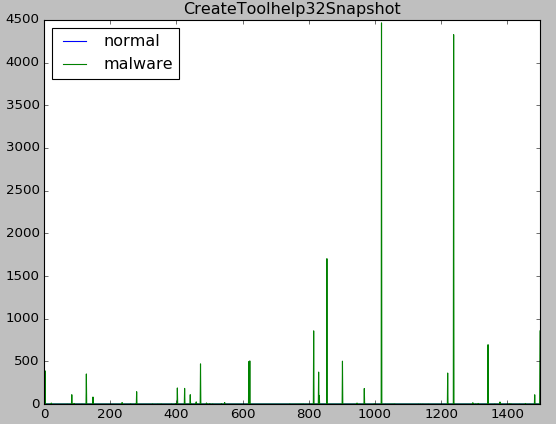
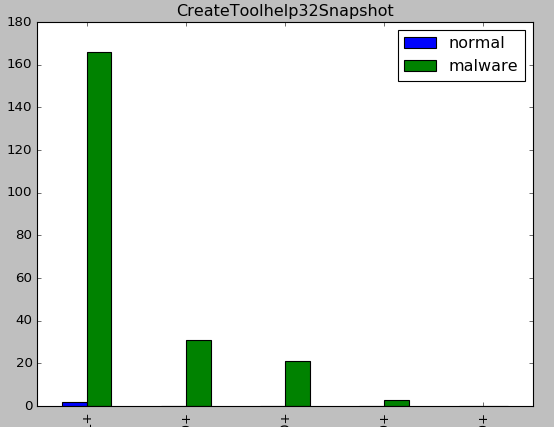


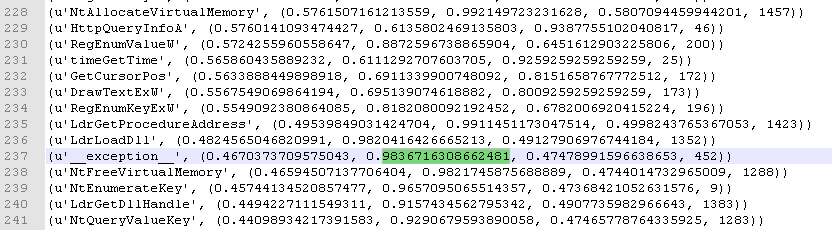
贡献值比较大的API基本都是病毒常用的，经常用来识别病毒。如排在前10的有：

WriteProcessMemory，NtGetContextThread，gethostbyname，CreateToolhelp32Snapshot，NtSetContextThread，Process32NextW，Process32FirstW，InternetOpenA，

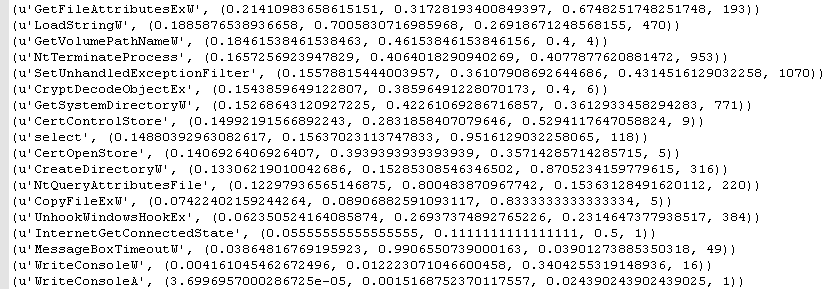
GetComputerNameA，GetFileSizeEx。其图像如下

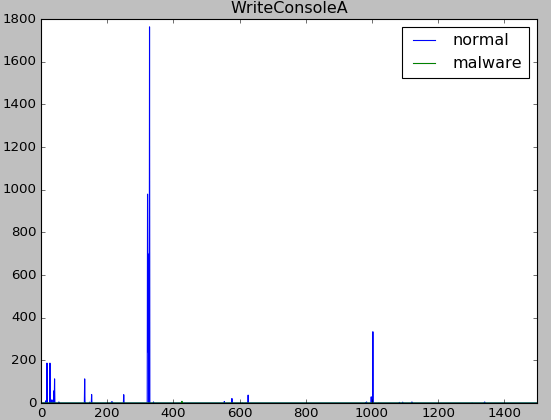
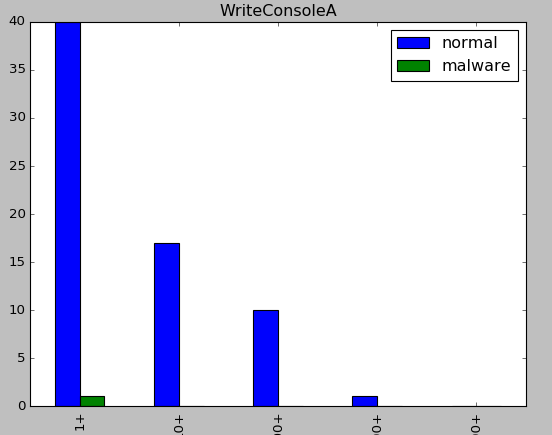
 

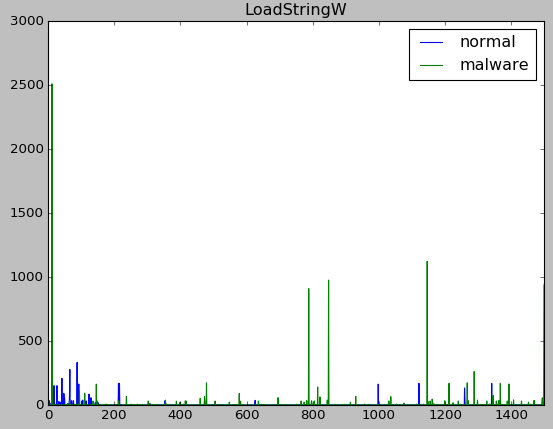
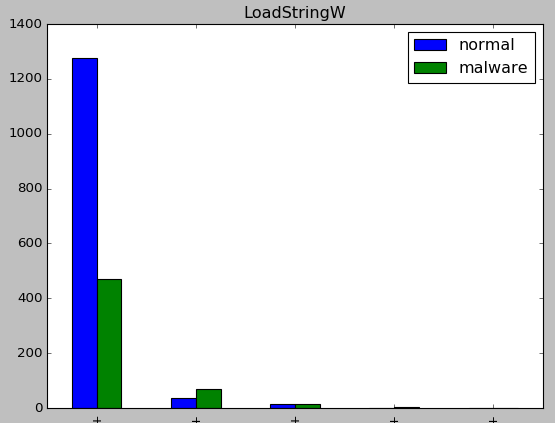


中间这部分数据用于下一节。



排在后面的都是比较没有病毒特性的API，比如WriteConsoleA，WriteConsoleW，MessageBoxTimeoutW，InternetGetConnectedState，select等。图像示例如下：

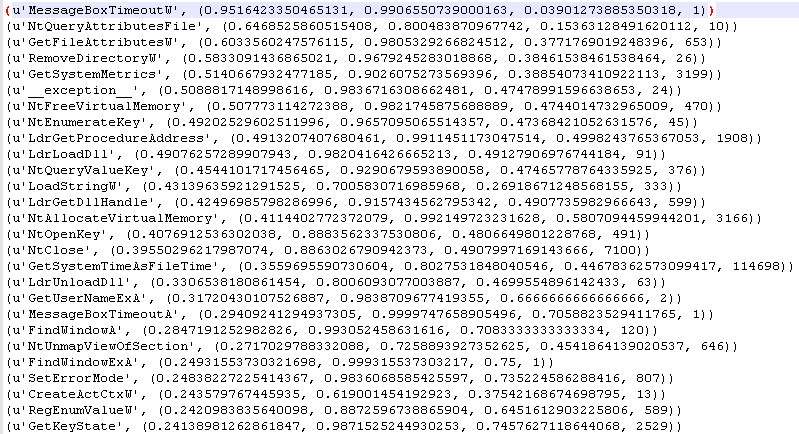
 

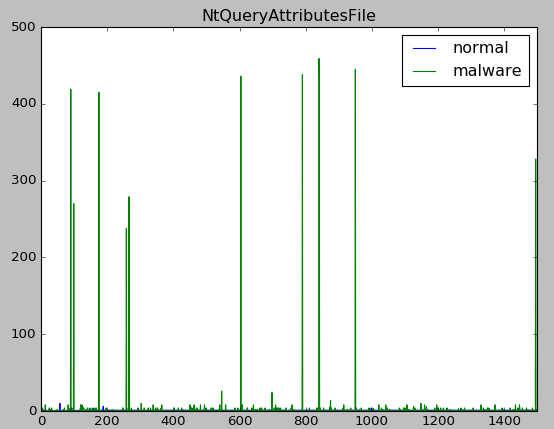
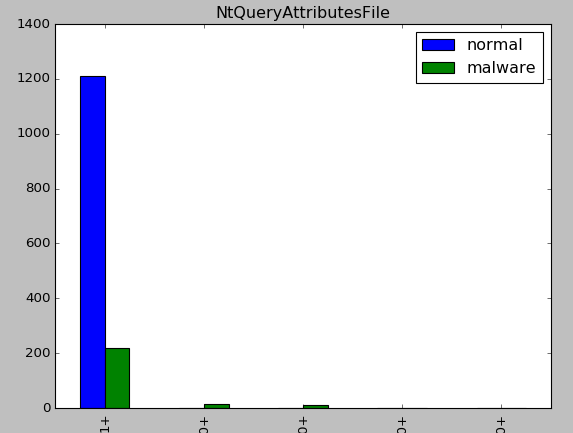
 

1. 计算P1-P2，进行排序，部分结果如下。

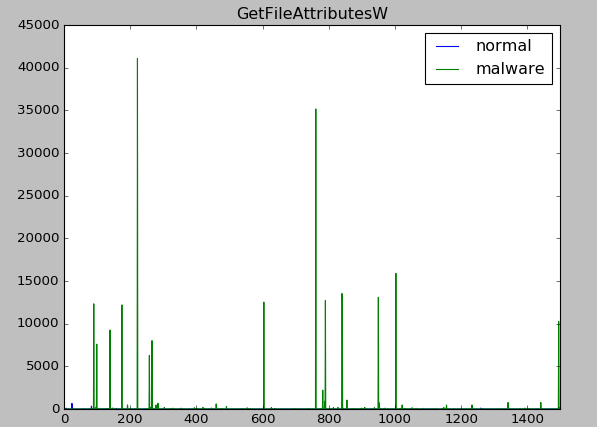
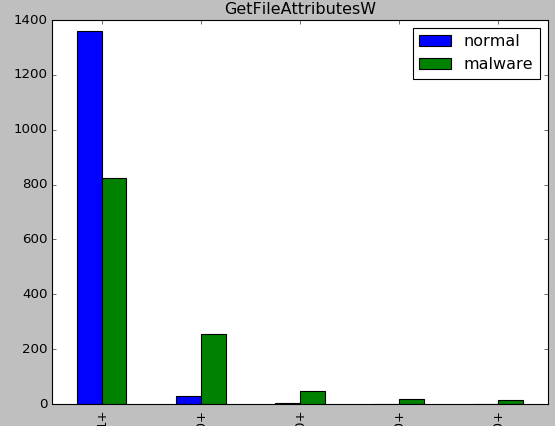
这部分可以找到那些低频次调用贡献度下，但是高频次调用贡献度大的API。

数据格式（API名,(提升度，API调用次数概率，调用API的样本为病毒的概率，正常样本调用最大次数)）

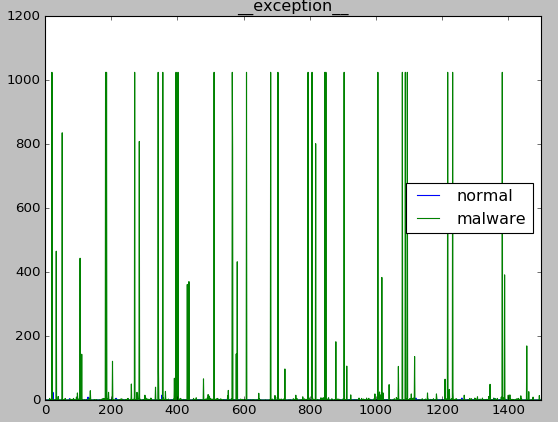
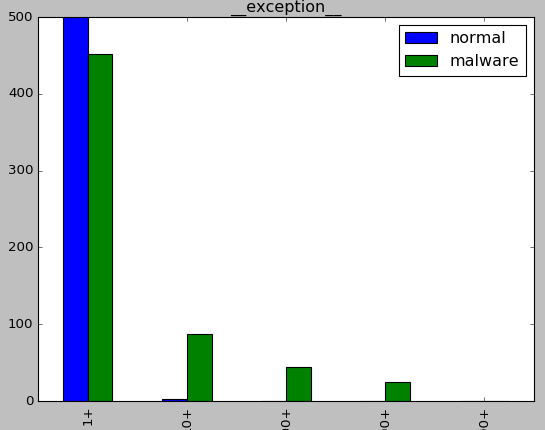


比如NtQueryAttributesFile，正常样本使用这个API的可能性更大。但是正常样本很少调用超过10次的。 因此超过10次调用NtQueryAttributesFile的样本就很可能是病毒了。

GetFileAttributesW 超过100次调用则很有可能是病毒。

超过10次的\_\_exception\_\_调用时病毒的概率很大。

完整的数据整理看数据部分。

## 结论

1. 通过计算贡献度=病毒调用API概率\*调用次数比例可以将API两极化

结果会受样本数量的影响，需要过滤掉次总数低的数据

2） 通过（调用次数比例-病毒调用API概率）可以找到那些调用次数少时对病毒分析没什么帮助，但是调用次数多时帮助很大的API。

## 数据

1. 对病毒识别影响大的API

|  |  |
| --- | --- |
| Top10概率为1的API | InternetOpenUrlA  NtReadVirtualMemory  CopyFileA  RtlDecompressBuffer  Module32NextW  Module32FirstW  OpenServiceA  CryptDecrypt  OutputDebugStringA  \_\_anomaly\_\_ |
| Top20概率高的API | WriteProcessMemory  NtGetContextThread  gethostbyname  CreateToolhelp32Snapshot  NtSetContextThread  Process32NextW  Process32FirstW  InternetOpenA  GetComputerNameA  GetFileSizeEx  MoveFileWithProgressW  NtWriteVirtualMemory  URLDownloadToFileW  CryptAcquireContextW  CopyFileW  WSASocketW  ControlService  NtOpenDirectoryObject  GetSystemWindowsDirectoryA  GetAsyncKeyState  DeleteService |

1. 调用次数少难以区分，调用次数多可以用于识别病毒的API

|  |  |
| --- | --- |
| Top20 高频调用特征明显的API | MessageBoxTimeoutW  NtQueryAttributesFile  GetFileAttributesW  RemoveDirectoryW  GetSystemMetrics',  \_\_exception\_\_  NtFreeVirtualMemory  NtEnumerateKey  LdrGetProcedureAddress  LdrLoadDll  NtQueryValueKey  LoadStringW  LdrGetDllHandle  NtAllocateVirtualMemory  NtOpenKey  NtClose  GetSystemTimeAsFileTime  LdrUnloadDll  GetUserNameExA  MessageBoxTimeoutA  FindWindowA |

详细数据见附件：

