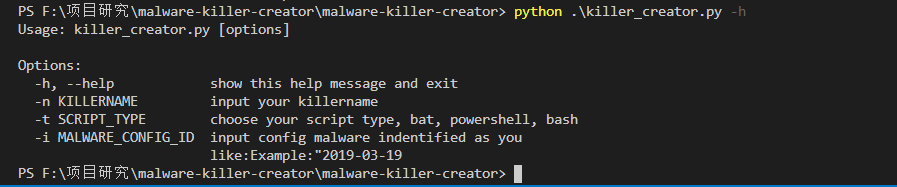
0x01--------------------前言

为了节省编写专杀的时间，并且为了更好的做好每一次专杀的记录，这里简单的写了一个工具来实现专杀的半自动化生成。



-n 指定你专杀脚本的名称

-t 指定你专杀脚本的类型

-i 指定你要根据哪个config的id来生成专杀脚本

0x02--------------------支持平台

这个工具目前只支持windows的bat, powershell生成。暂时还不支持linux的bash, 而且对于linux平台，使用bash还是直接使用python现在还在考虑当中。其中bat只支持简单的文件，注册表，进程，服务，计划任务查杀(主要是我不想在花一些精力在bat上了)。Powershell可以支持更加精确的查杀，可以避免一些误杀。详情见下文。配置文件的格式采用json格式，简单，快捷，识别快。

0x03--------------------bat配置解析。



bat只支持, 注册表键值、键删除，服务根据名称删除(非DisplayName)，进程根据进程命删除，计划任务根据计划任务名称删除, 文件根据文件名称删除。

注册表:

1、当删除某个键下面的某个值时，采用key:value的格式来写。

2、当删除某个键时，采用key:0的格式，后门的value必须为0

服务：

1、因为根据字典的规则，key不能相同，所以我们采取”name\_xxx”:”服务名称”的格式，必须包含name\_, 会自动识别。

进程：

1、进程的配置格式和服务的类似。

计划任务：

1、计划任务的格式和服务类似，不过以taskname开头的任意”taskname\_xxx”就行。

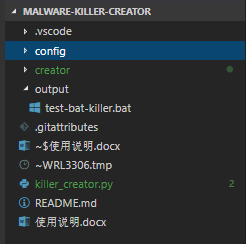
文件：

1、文件的配置格式和服务的类似。

Example:



输出的专杀bat文件会写到output目录下。



生成的bat文件。

注意：我不建议使用bat。对于必须使用bat的系统。无话可说。

0x04-----------------------powershell配置格式解析



Powershell支持更加精准的查杀和模糊查杀。

注册表：

注册表和bat的配置格式一样

服务：

1、服务支持根据服务名称查杀，名称配置格式采取”name\_xxx”:”服务名称”的格式

2、大部分服务是使用exe格式的服务程序来实现的，所以我们可以根据服务程序的sha256的hash来进行精准查杀。使用hash查杀的采取”hash\_xxxx”:” sha256hash”格式

3、服务还支持dll格式的服务程序，这里我们就需要检测注册表了并且，根据注册表的内容和文件的hash来查杀(此功能暂时未实现，等有时间再说。)

进程：

1、首先就是普通的进程名称匹配查杀。名称格式采取“name\_xxx”:”进程名称”。

2、对进程的文件的sha256 hash进程匹配查杀。名称格式采取”hash\_xxx”:”sha256 hash”

3、还有一些恶意进程，比如，powershell等，恶意内容主要是在命令行这里，支持对进程的命令行模糊匹配删除

计划任务：

1、计划任务支持普通的计划任务名查杀。名称格式采取”namepath\_xxx”:”计划任务名称”

2、因为有些计划任务是使用了powershell和参数组合，所以支持命令行模糊匹配查杀。

格式采取“command\_xxx”:”命令行正则”

3、计划任务也有直接执行可执行程序的，我们可以根据计划任务的计算可执行程序的sha256的hash来进行匹配，实现计划任务精准查杀。格式”task\_execute\_hash\_xxx”:”sha256hash”

文件：

1、文件支持直接文件名称查杀。格式”name\_xxx”:”文件路径”

2、文件支持根据文件的路径和hash查杀。格式“hash\_数字\_文件路径”:”sha256 hash”，这里的数字这里算是一个bug吧，后续修改。

3、本来还想支持根据文件的内容模糊查杀的(等有时间再写吧)

wmi:

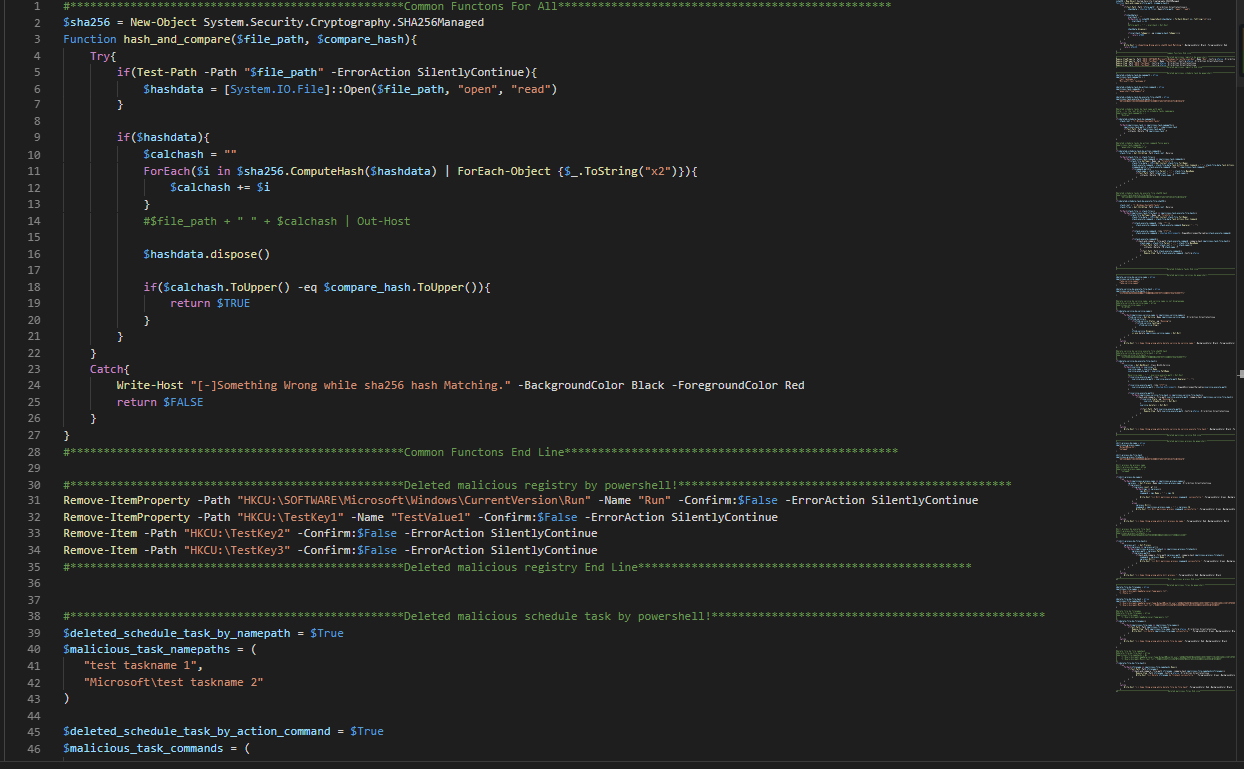
1、支持根据wmi的eventfilter的name来删除。格式”name\_xxx”:”filter\_name”。

2、支持根据wmi的eventconsumer的commandline来模糊匹配删除。格式”commandline\_xxx”:”consumer\_commandline”。

注:其实wmi我是不想写的，因为可以使用autoruns等工具来进行删除。

Example:





0x05-----------------------工具更新

在这里首先说一下，为啥要这么多”xxx”, 因为使用的是json的格式，字典中key不唯一的话，是会报错的。

将来的计划任务，慢慢完善，除了上面的未实现的功能，我好像还忘记写了清除wmi的。