其实所谓的杀木马是假的，只是把疑似被感染的文件删除而已

特征码有两部分：真正的特征码+MD5校验码

那部分真正的特征码是原来开元的东西

MD5值是我们自己添加的

例如我们发现有个文件，a.exe是疑似木马文件，就用别的软件提取它的MD5值，然后就加入到库中

哪个文件你可以用txt文件打开，哪个文件你可以用txt文件打开

真正核心的东西应该是扫描引擎，而这部分是我们不知道的，都是拿来开源的东西用的，人家开源的这个引擎其实已经封装成dll了，也不需要我们自己封装

就是这个引擎从被扫描的文件中提取特征码，与库中的特征码匹配。

在我们自己写的源代码中你可以发现其实使用了这些接口来操作这个引擎的

仅仅使用了几个最简单的，因为复杂的我们还不敢用，不知道怎么用

我记得cl\_engine\_new是用过的

Clamav

还有，其实根本没有所谓的行为特征库，或者至少，这个所谓的行为特征库根本没啥实质内容

他们所谓的行为特征库的意思是：因为有了文件过滤驱动，所以可以监控到大部分对文件系统的操作，例如写文件、创建文件、读文件等，那么，我们自己臆想一下：如果首先发生了读文件、又发生了写文件，我们就把他当做是病毒了

扫描的时候肯定只是静态的。动态的监控实际是需要当有害行为发生时才有可能报告的。

但是应该知道，真正的病毒的行为特征库不是这么简单一两句话可以总结的，所以也不敢把行为特征做成库，否则很容易把正常软件误报

那个网络驱动其实也是一个过滤驱动，即filter driver。它是放在网卡驱动之上的一个驱动程序，原来我们的网卡驱动在没有这个filter driver之前，所有我们电脑发送和接收的网络数据包都是由网卡驱动来做的，也就是说，网卡驱动控制着网卡，告诉网卡计算机要发送什么数据包，要接收哪些数据包，然后网卡就会照办，

现在在网卡驱动之上有了过滤驱动之后，我们计算机要接收的网络数据包会传送到我们写的filter dirver里面，然后我们就可以自己在filter driver中解析这个数据包，看看ip之类的，如果是我们不想要接收的ip，那么我们的filter driver就可以把这个数据包丢掉，不把它发送给我们的应用层程序，这样我们实际就收不到某个ip的数据了，我们写的网络过滤驱动就是这么个原理，over。

实际filter可以做的工作有很多，因为它已经接收了所有的网络数据包，那么解析报文的结果，不仅仅能看到ip地址，还可以解析网络协议、解析出报文的内容，例如传送的是“狗屎”还是别的，那么就可以根据这些来进一步过滤报文了，但是因为复杂，这也涉及更深的知识，特别是算法一类的，所以也做不出来

驱动的分类：bus driver\function driver\filter driver，文件过滤驱动实际上是挂在文件系统驱动（file system driver）之上的过滤驱动，插件扫描、恶意代码检测、枚举系统服务、开机启动项等，

这个软件能判断出来是不是木马吗？其实能部分判断出来，这只是开源引擎的功劳，但是我们在做的时候，把人家的特征库砍掉了90%，所以其实没那么准确了，我们的功劳啊。。。。就是拼凑了一下，然后写了一些无关紧要的小功能，他把摆渡木马的检测当成普通木马的检测了，所以根本不关摆渡木马什么事儿，

好比是说，我们要找汗血宝马，但是我们派去找马的人，只会认普通的马，于是他就说：汗血宝马也是马，认识马那么就一定能把汗血宝马也找出来

所以，只是说最后找到了100匹马，其中可能有1匹汗血宝马，也可能没有。他还要邀功说：看我派的人多优秀，不光找到了汗血宝马，连所有的马都找出来了，但实际上，人家摆渡木马根本不是这么查的，而且这么查可能查到的是普通的木马，根本查不到摆渡木马，所以说，不关摆渡木马任何事儿，而且跟安全有关的事儿也不是我们自己做的，而且还用不好人家的引擎，还阉割了人家的引擎，