高级Windows2000Rootkit检测技术(2) PDF转换可能丢失图片或格式，建议阅读原文
https://www.100test.com/kao_ti2020/140/2021_2022__E9_AB_98_E7_BA_A7Wind_c100_140410.htm 有一个明显的频率高点。如图4和5中所表示那样的，即使是在系统负载很大时，频率高点所对应的数值保持不变。很难解释这个令人吃惊的现象，可能是因为在循环中同一个系统服务被调用几百次后，与这个系统服务相关的缓冲最后会被填入固定的值。 假想现在有人安装了隐藏文件的rootkit，如果我们重复测试并绘制相应的条形图，就会发现频率高点向右移了，这是因为rootkit需要进行隐藏文件的工作。 在现在的代码实现中，只进行了少量的测试，包括典型的服务如：文件系统读取，枚举进程，枚举注册表项以及socket读取。 这些测试将有效地检测出著名的ntrootkit(见[1])，或最近比较流行的hacker defender(见[4])，包括它自带的网络后门，当然还包括很多其他的后门。但要检测出一些更好的后门还要加入一些新的测试。 误报和执行路径跟踪 虽然对频率高点的检测有助于我们处理系统的不确定因素，但有时会发现测试得到的值有小的差值，一般来说不大于20。 有时这会是一个很严重的问题，因为我们不能确定那些多出来的指令意味着被入侵或只是正常的误差。 为解决这个问题，我们使用了执行路径记录模式。和单一的epa模式比较，系统增加了对执行路径的记录(包括地址和运行的指令)，首先，系统记录下正常情况下的执行路径，以后的每一次运行将产生diff文件(正常系统和现行系统之间的比较)。我们应该使用好的反编译器来分析那些不一样的地方，以此判定他们是否可疑。图6是一个diff文件的例子。 现阶段的diff

文件只记录下指令的地址，以后可能将两次测试的不同结果存为pe格式文件，并可用 检测 ”offset-in-the-code ” 的变化 想象有这样一个rootkit，它基本和上面提到的 fu rootkit (见[3]) 一样，但不从psactiveprocesslist中，而是从分派器使用的数据结构中移除进程。我说过那不可能，因为隐藏的进程将分配不到运行时间..... 然而，rootkit可以同时更改分派器代码中所使用数据结构的地址(offset)，换句话说，就是使其使用不同的链表。但只有分派器使用这个 ” 新的 ” 链表，而系统其他地方还是使用 ” 旧的 ” 链表..... (见图7)。 虽然这种技术不会改变执行指令的个数，我们还是能检测到它，但需要进一步的完善现有的工具。这项功能现在还没有实现，但应该不是很难。 针对epa的攻防 我们可以想到一些能骗过epa类检测工具的方法，先把它们分为两类。 1、针对特定工具的欺骗 2、对epa类技术的通用攻击 首先，我们考虑一下通用的攻击方法和怎样防止这类攻击。接着讨论针对特定工具的攻击以及怎样通过多态来预防。 对epa类技术的通用攻击