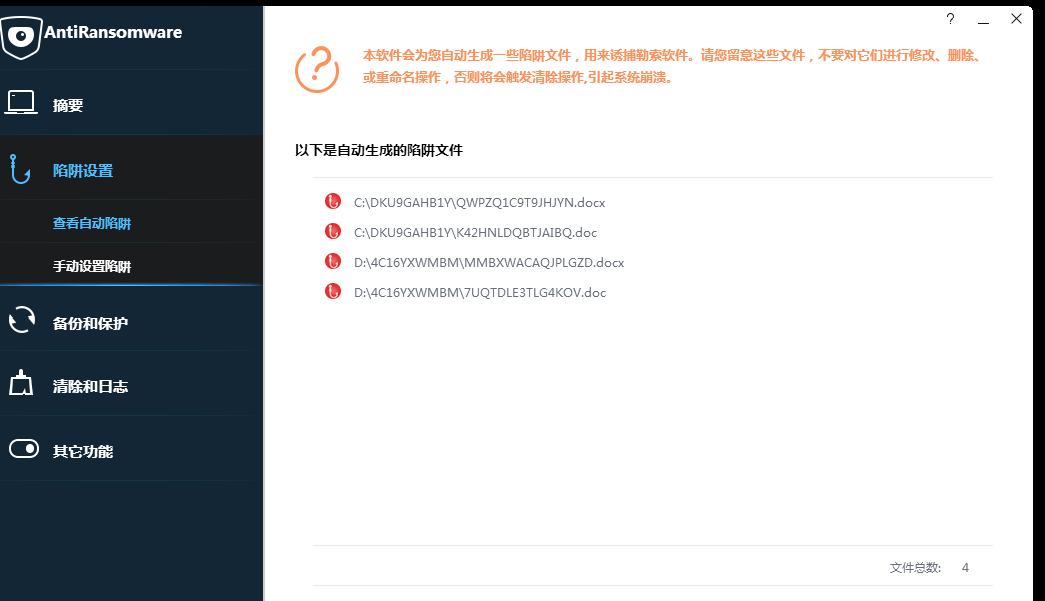
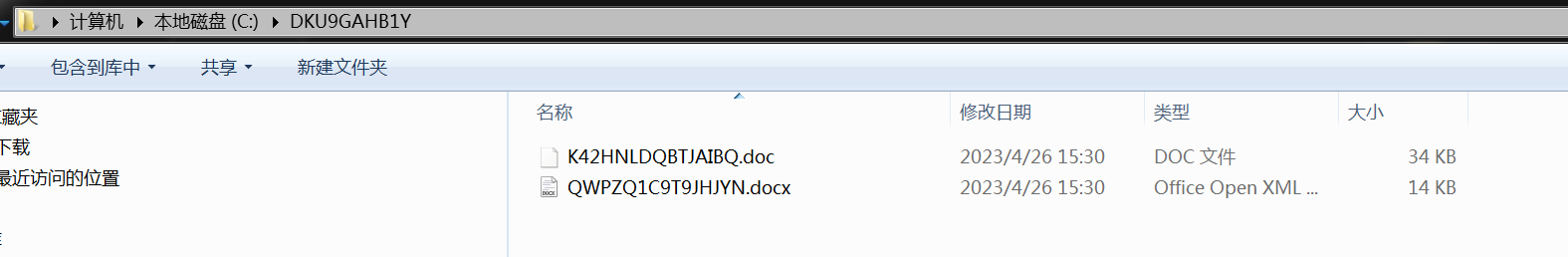
**AntiRansomware Notes**



自动诱饵文件系统截图



C盘自动诱饵文件截图

用户手动设置诱饵文件，本次将桌面的一个test文件成功拖拽进入界面中，直接显示了其文件路径，功能测试成功。

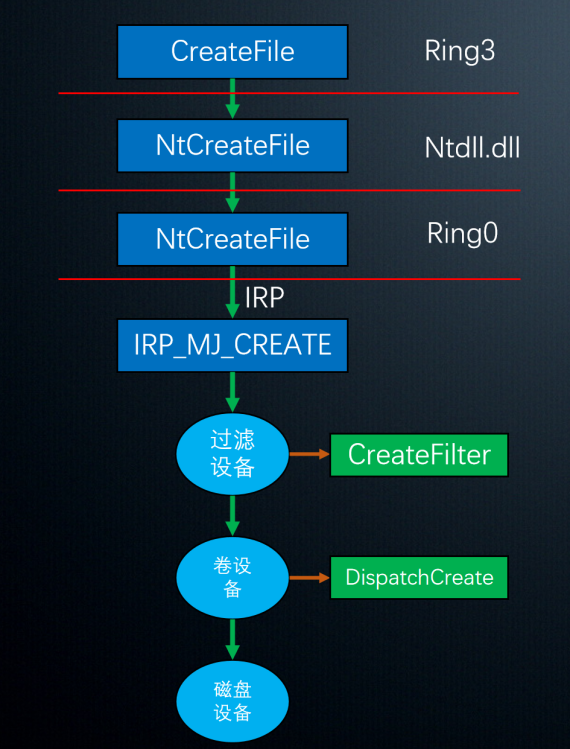


手动诱饵文件系统截图

驱动模块流程

SFilter框架流程

Windows结构是分为ring3-ring0（用户层-内核层）结构，以createfile这个api为例子，该api为ring3层，通过不断追溯底层调用可以到ring0层NtCreateFile，本系统的SFilter文件过滤框架是在Ring0层与虚拟设备之间连接，如图中过滤设备就是本设计的文件过滤框架所在之处，需要和卷设备进行绑定，而卷设备又对应了主机的磁盘，这样的逻辑联系就是整体的框架。消息优先通过过滤设备然后通过卷设备与磁盘设备。

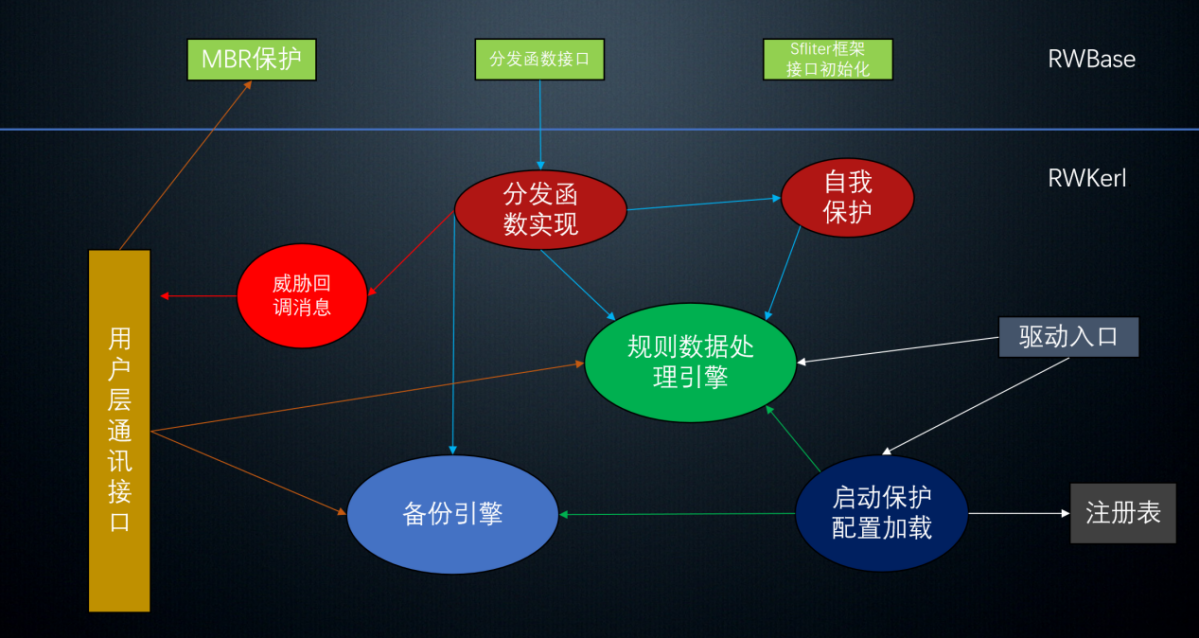


SFilter框架图



过滤设备-卷设备-磁盘设备关系图

驱动模块框架



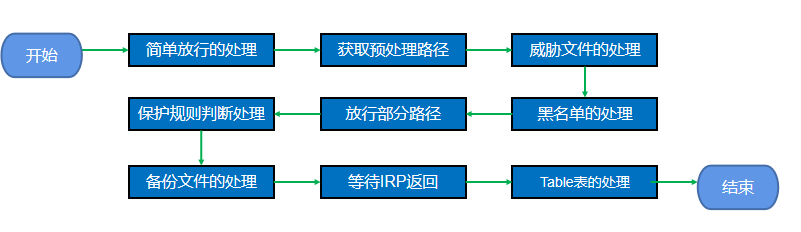
RWBASE-RWKERL关系图

本设计的驱动模块分为两个部分，分别为RWBase（基础模块）与RWKerl（业务模块），基础模块提供基础的过滤驱动框架以及分发函数接口，而实际的部分是在业务模块中实现，例如分发函数的实现、规则处理、配置加载、备份引擎等等备份。这样的框架与业务分离的设计有助于未来的功能维护和升级，开发者可以只在业务模块进行功能的编写，不用考虑基础的一些模块与框架的，减少开发成本与时间。

分发函数设计流程

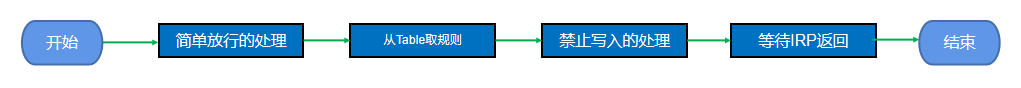
分发函数的设计是驱动模块中的关键，相当于规则来约束最终的文件过滤实现。一共有4个分发函数Create,Write,SetInformation,DirectoryControl。

Create，首先获取一些预处理配置文件路径，接着对重启后锁定清除威胁文件的处理，接着是对黑名单的处理，放行一些系统资源的路径（因为系统本身就会对这些文件夹频繁的读取，且勒索软件不会对这些文件夹读取，所以选择放行），然后对保护规则进行判断处理，是否对在保护目录的大文件进行读写操作，接着是对备份的文件进行处理，随后等待下层设备IRP，最后在到Table表中存放。另外，Table表是所有分发函数共同使用的表，存放文件对象等内容。



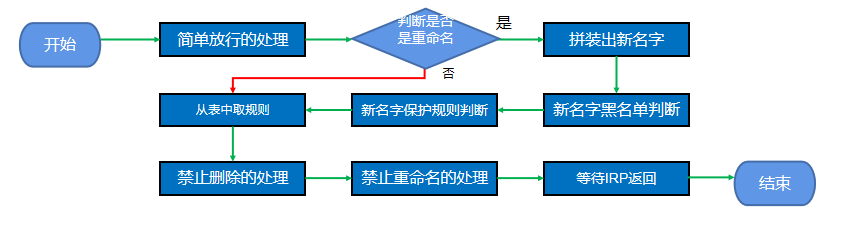
Create分发函数流程图

Write，write管理文件写操作，从表中获取规则然后，判断并禁止写入的操作，然后等待下层设备的IRP返回。



Write分发函数流程图

SetInformation，该分发函数是对重命名的操作处理，首先判断是否进行了重命名操作，没有则直接进行表中处理，若确实是重命名操作接着先预处理拼接出新名称对其进行黑名单判断、保护规则判断，若均不存在那么继续继续table表取对象然后对其进行禁止删除、重命名操作，最后等待下层设备IRP返回。



SetInformation分发函数流程图

DirectoryControl，该分发函数实现的是文件路径的隐藏功能，前文的备份路径隐藏的实现是通过该分发函数在RING0层实现的。

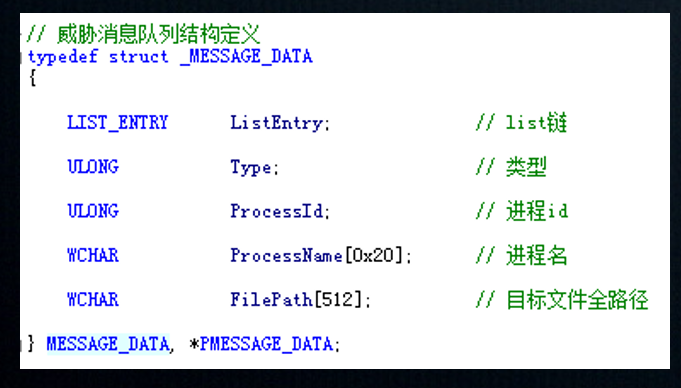


DirectoryControl分发函数流程图

威胁消息处理流程

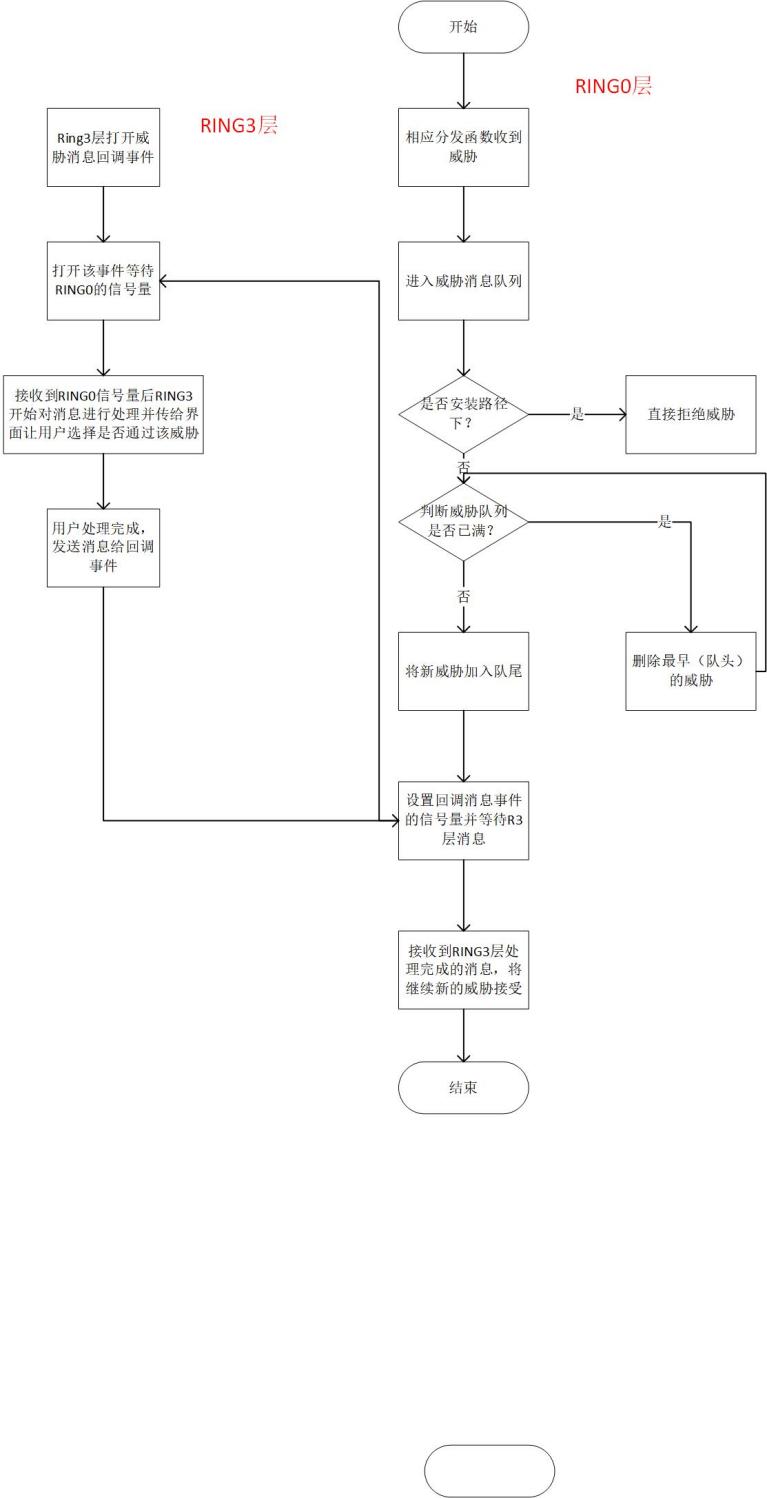
威胁消息处理是在RING0层处理完成发送给RING3层用户界面的弹窗提示之间的整个过程。

威胁消息有威胁消息队列来存放，记录了威胁进程的进程名称和ID以及威胁文件路径。



威胁消息队列结构定义

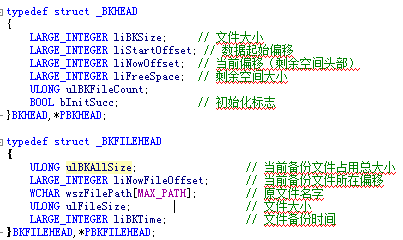
从分发函数截取的符合威胁条件的疑似威胁文件传入威胁队列中，首先判断威胁是否来自于本系统的安装目录下，如果是则直接阻止不传入威胁队列。接着判断威胁队列是否装满，如果装满那么就把队列头部最早的威胁进行删除，然后将新威胁传入对队尾，设置回调消息的信号量并且等待RING3层的回复消息。这样RING3层的等待信号接收到信号量后对消息进行相对应的处理，把消息传送给用户界面进行威胁弹窗让用户选择是否放行或处理，最后用户处理完成，RING3通过回调事件发送信号量给RING0的等待事件，RING0处理完成事件后将继续处理新的威胁，以此往复。



威胁消息RING0-RING3层处理流程

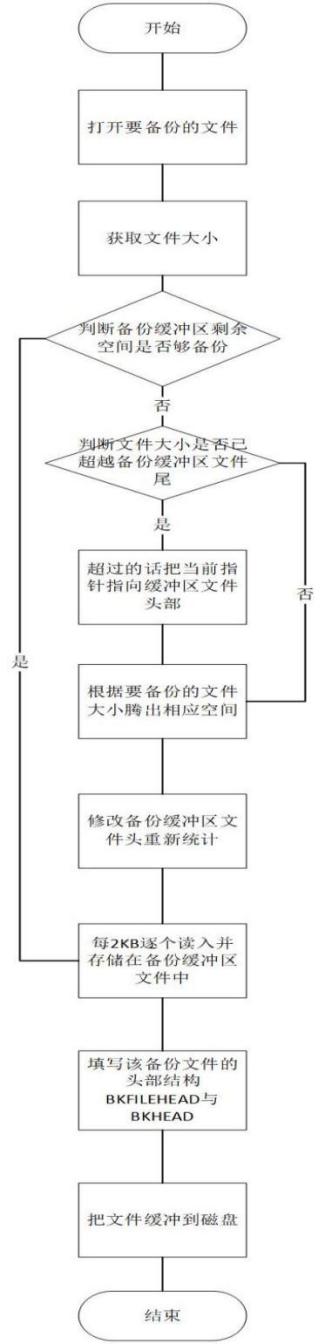
备份操作处理流程

本设计的备份模块采用的是循环备份，在结构上是如果备份队列填满以后那么将会从头开始覆盖式的重新填入需要的备份内容，这样往复循环的进行备份操作。



备份队列结构

备份核心逻辑思路为打开要备份文件并获取其文件大小，然后判断备份缓冲区剩余容量是否足够备份该文件，1）如果无法满足那么就判断文件大小能否满足备份缓冲区文件末尾，先填充满末尾，接着把当前指针从末尾指向缓冲区文件头部，然后根据备份文件的大小腾出相应空间，修改缓存区文件头（该文件头数值为数据起始偏移值）。接着进行2）的操作。2）每2KB逐步读入到备份缓冲区中，然后填入备份文件头部（BKFILEDHEAD)结构以及修改缓存区文件头，最后把文件缓冲区更新到磁盘中完成备份操作。



备份处理流程图

第三个功能为备份与保护功能，备份设置子界面可以查看当前选择的备份磁盘以及其他磁盘的信息，用户可以在此界面进行备份磁盘的更改，同时可以查看初始化系统时候划分给备份的磁盘空间大小，另外最下面可以进行文件备份阈值的修改，通过点击需要的阈值（25M-300M)进行修改。



小文件

大文件

备份配置系统截图

用户可以在此界面点击左下方的添加文件功能，然后拖拽文件进入到列表区域，这样即可针对进入的文件进行文件保护功能，同样也可以删除已经存在保护的文件（修改文件同理触发保护），同时如果列表内文件数量庞大本功能也支持模糊查询。



大文件保护系统截图

在这里可以查看已经在规则库中的文件扩展名，同时用户可以对备份类型进行删除以及添加自定义的扩展名，所有符合备份类型且大小在备份上限内的文件在修改、重命名、删除时候都将对其进行备份到备份目录中。

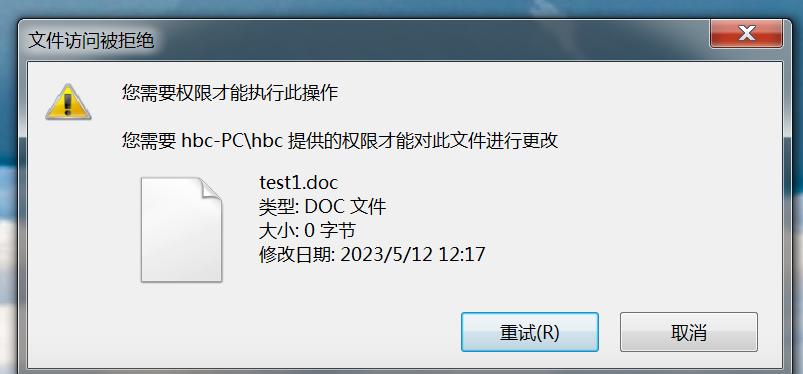


备份类型系统截图

用test文件进行文件保护测试，当我们对保护中的test采取删除操作时候，防御系统将进行威胁弹窗提示用户是否对该操作进行放行，如果点击取消则阻断该操作，如果需要放行那么需要输入系统密码然后点击确实按钮。



文件操作警告弹窗系统截图

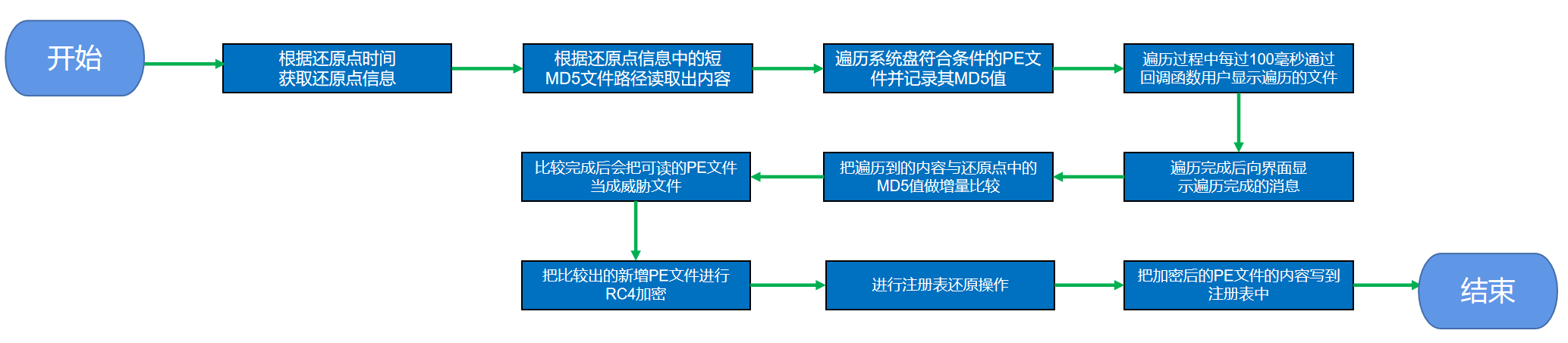


测试样例截图

还原清除流程

还原计算的运行流程

通过还原点时间对应着还原点的信息，还原点时间相当于一个识别ID，然后根据还原点信息中的MD5后的文件路径读取出内容，接着遍历磁盘寻找符合条件的PE文件并记录MD5（该过程中不断通过回调函数同步给用户界面查看进度）。然后比较遍历md5值与还原点md5值，把比较出的新增疑似威胁文件PE写道文件中并对其进行RC4加密（防止其继续对外威胁以及防止被其他进程消除），继续进行注册表还原操作，把加密后的疑似文件写入到注册表中。



还原点计算流程图

为还原清除，手动清除部分可以查看已经存在的还原点创建时间，左下方可以设置还原点生成参数，比如还原点生成间隔时间，还原点保存个数上限（考虑到系统存储资源）。

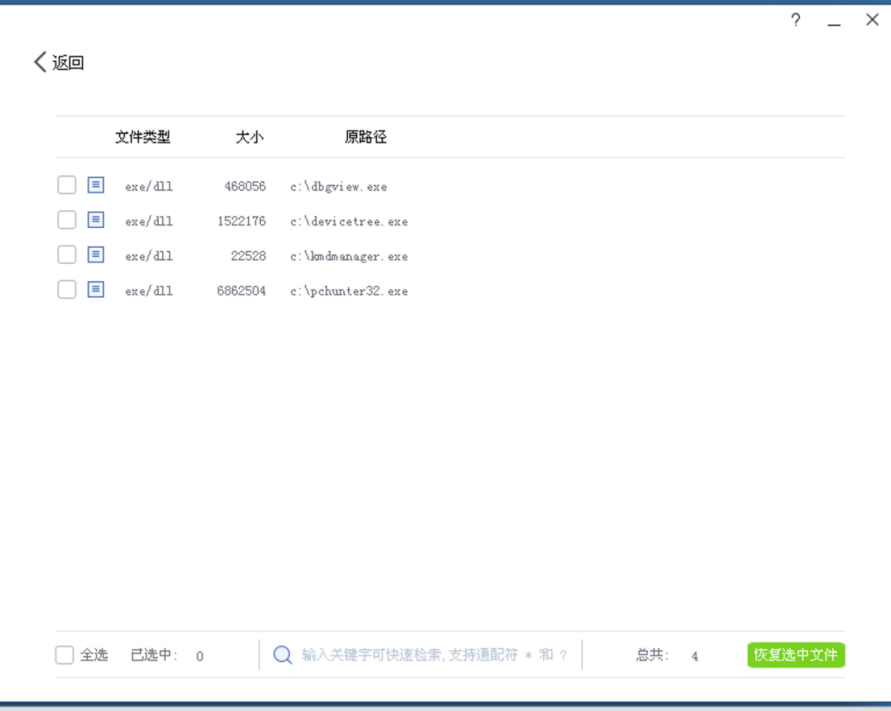


还原点系统截图



还原点生成配置截图

所有的清除日志列表中的内容，包括清除时间，文件总大小与个数，点击查看按钮可以查看每次清除的所有文件，同时可以看到所有的疑似威胁被删除的文件，用户可以勾选其中系统误删的文件进行恢复操作。



清除日志列表截图

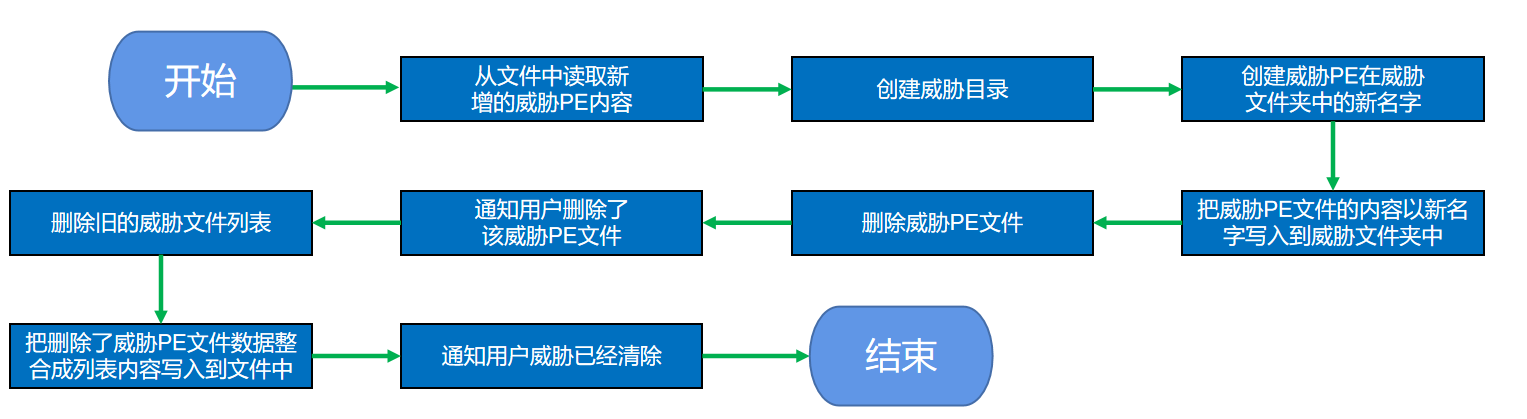
接下来是测试一下威胁入侵时候的还原清除的流程，当触发了陷阱文件以后将弹出威胁警告弹窗，显示出咸鱼进程名字和进程号，同时提示用户及时保存手头工作内容文件到唯一可以写入的文件夹。然后选择已经存在的还原点进行还原清除流程。



还原点触发警告截图

重启后的清除流程

系统重启完成后，创建威胁目录，并从文件中读取新增的PE文件内容放到目录中并对其进行重命名，接着通知用户删除了威胁PE文件，然后删除威胁文件列表，同时同步刚刚删除的威胁文件PE列表到界面让用户可以查看并恢复误删的文件。



重启后清除流程图

接着进行四个弹窗流程，生成文件快照，通过对比锁定出疑似威胁的PE文件，然后进行还原注册表随后主机系统进行重启。



还原过程截图

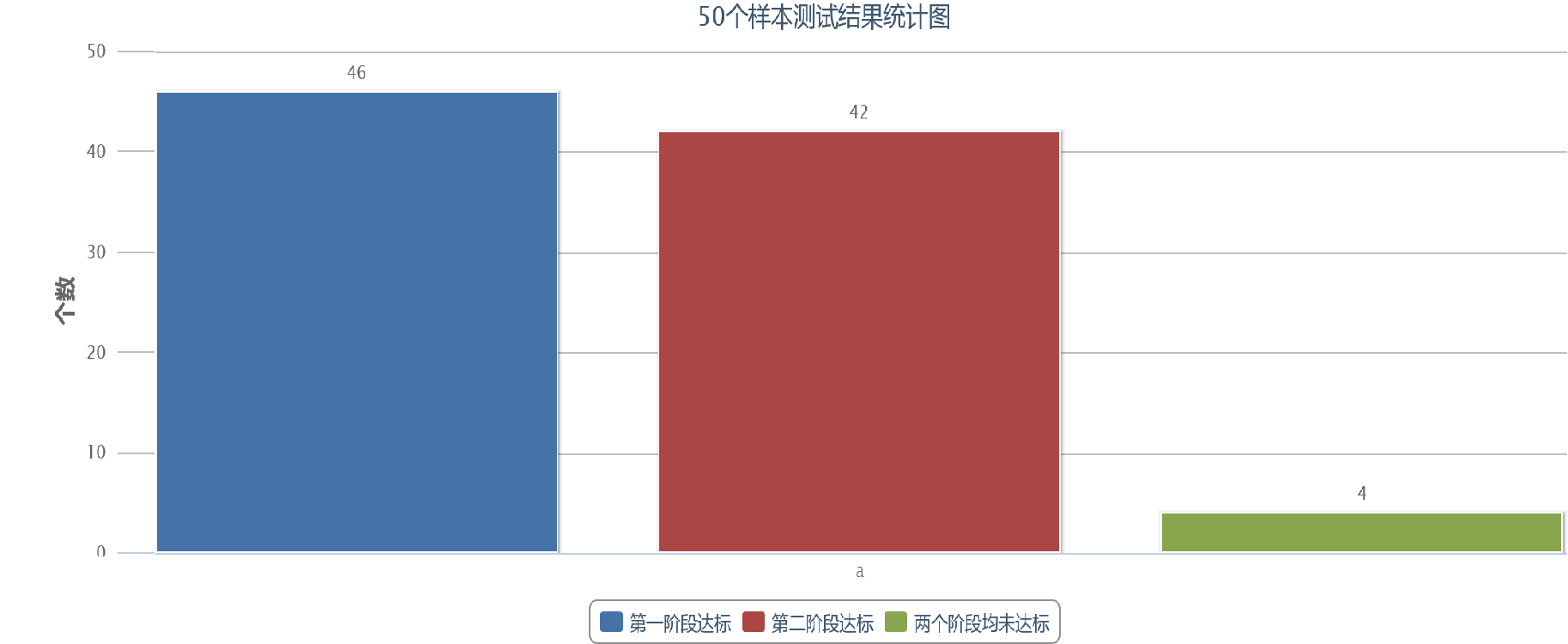
重启以后，系统弹窗提示清除完备的所有文件路径，用户可以到清除日志中恢复误删的文件。



威胁文件清除截图

系统防御效果

把防御分为两个程度，防御程度1为能够顺利检测出勒索软件然后回到界面提示用户。防御程度2在1的程度上可以进行阻断操作并能够在重启以后进行清除威胁文件功能。达标为满足防御程度1或防御程度2的要求，未达标为未满足防御程度1或防御程度2的要求。



50个样本测试结果统计图

对50个样本进行测试，通过统计有46个第一阶段达标（92%），有42个第二阶段达标（84%），有4个两个阶段都不达标（8%）。

部分样例测试表

|  |  |  |
| --- | --- | --- |
| 测试勒索软件名称 | 防御程度1 | 防御程度2 |
| LockBit1 | 达标 | 达标 |
| Cerber15 | 达标 | 达标 |
| CryptoShield\_2 | 达标 | 达标 |
| CryptoWire\_9 | 未达标 | 未达标 |
| Gandcrab\_12 | 达标 | 达标 |
| Maze\_11 | 达标 | 达标 |

|  |  |  |
| --- | --- | --- |
| Crysis14 | 达标 | 达标 |
| MaMoCrypter | 未达标 | 未达标 |
| Ransomware.FRS\_2 | 达标 | 达标 |
| Dharma\_10 | 达标 | 未达标 |

测试结果分析

部分测试样例其中2款（CryptoWire\_9与MaMoCrypter）未达到防御程度1的勒索软件样本，可以发现二者的共同点是都未对诱饵文件进行操作，其中CryptoWire\_9只对桌面文件进行了加密。另外，二者的后缀名也不在黑名单中，这二点可能是导致检测模块无法检测出它们从而无法完成后面的还原清除操作，而只能达到防御程度1。还有1款（Dharma\_10）未达到防御程度2的标准，它可以顺利被检测但是在还原清除环节无法进行，那么据此可以推测该勒索软件被检测模块顺利检测出但是在还原清除重启系统以后，勒索软件的启动优先级高于本系统从而优先于勒索软件清除，最后对文件进行了加密。