Windows 自启动项的查看与分析

姓名：刘浩文　学号：517021911065　邮箱：IssacLiewX@sjtu.edu.cn

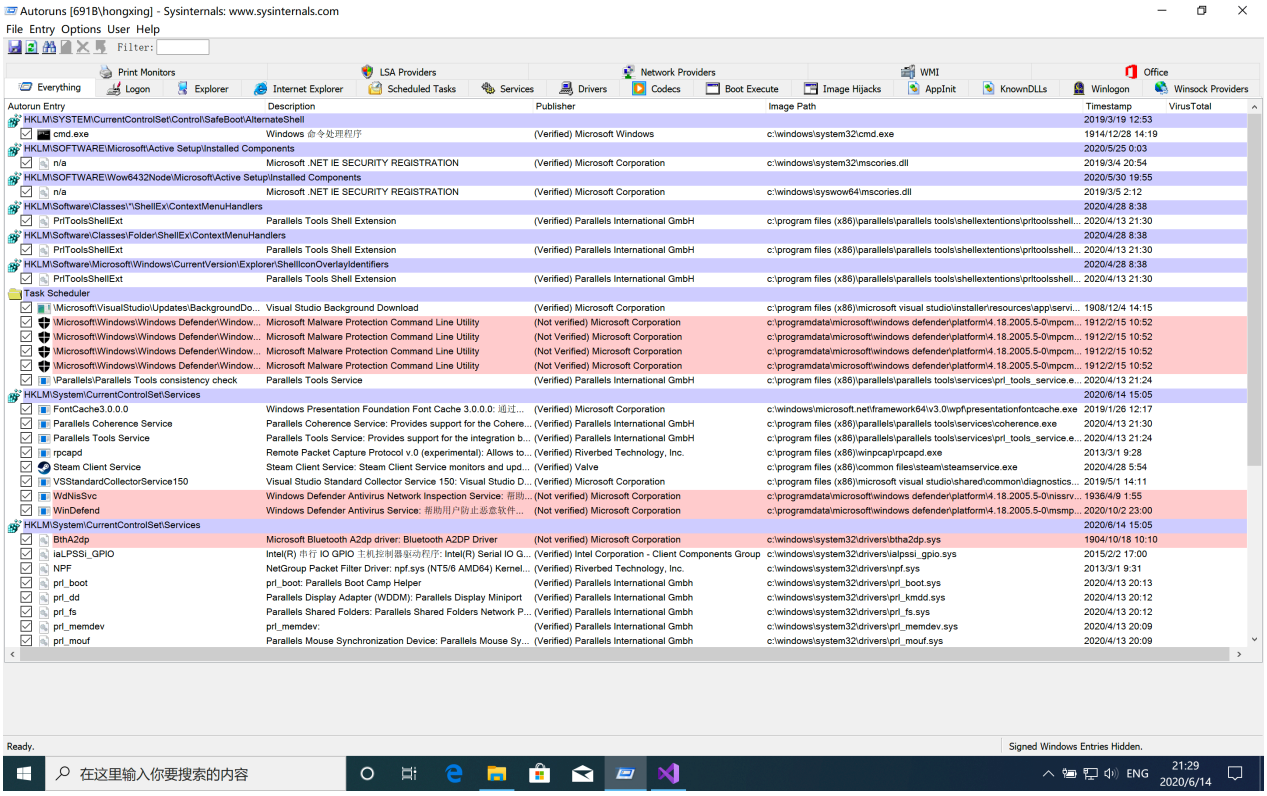
Windows 自启动项

为什么会有 **Windows 自启动项**？一些系统进程，如一些维护系统安全的进程与一些硬件驱动程序，需要在系统开机时就自动启动，否则会影响系统的启动或损害系统的功能。所以这些进程有必要设置成自启动。还有一些没有必要自启动的进程，但如果将其设置为自启动将会方便一些用户使用计算机功能，所以也可以将它们设置为自启动。

Windows 自动启动功能在给用户带来便捷的同时，也给病毒提供了便利的启动途径。所以无论是出于方便自己使用，还是防范病毒感染电脑，都有必要深挖一下 **windows** 中所有的启动项。

从 Autoruns 软件开始探索

Autoruns是一款功能强大的管理开机自启动项目组的软件。

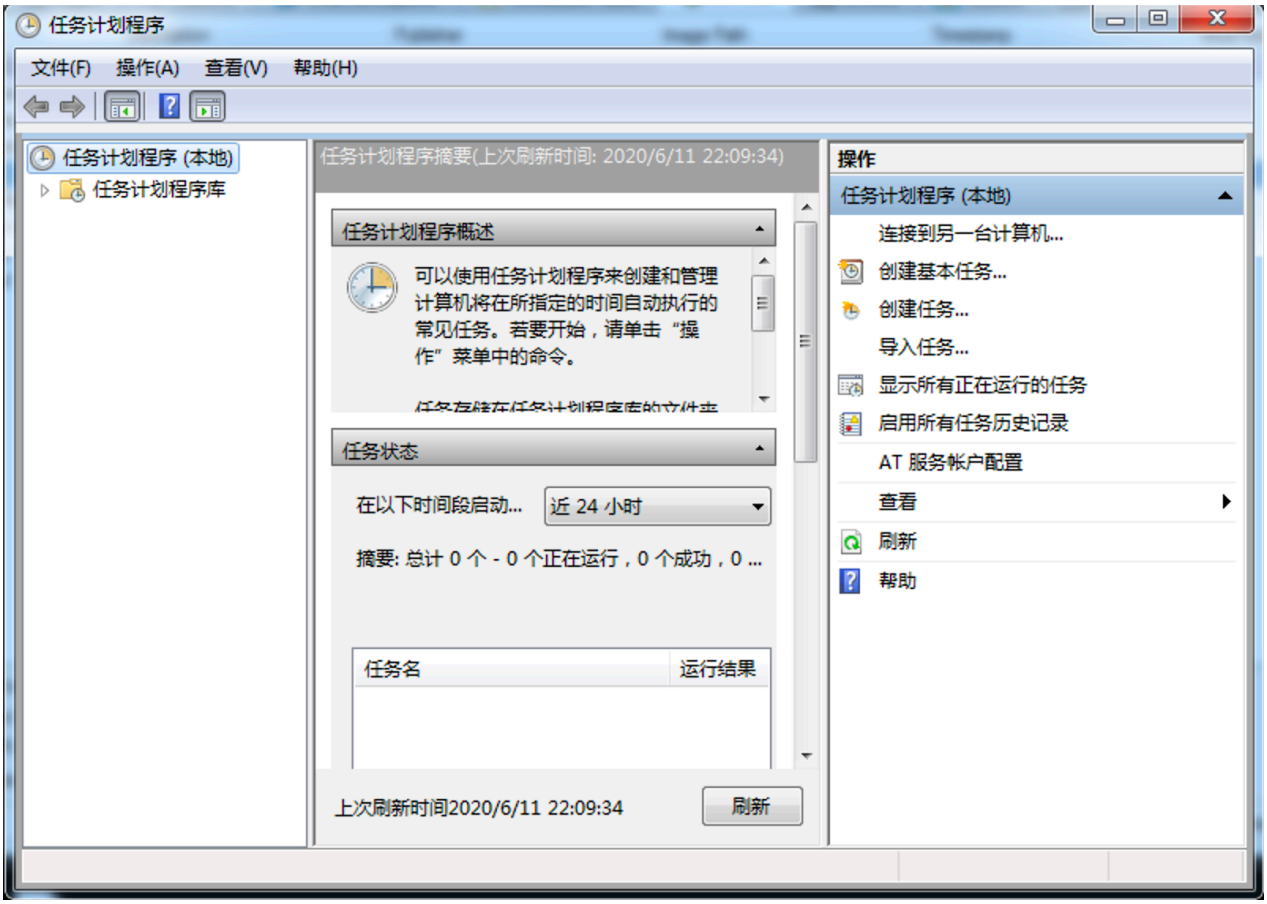


Everything：全部的开机自启动项都在这个标签下。

Logon：*LOGON* 进程是登录的初始者，也是用户在使用安装有 *NT* 计算机时通常会经历的登录阶段的前台程序。LOGON本身是个很特殊的应用程序。名字为登录，但是实际上并不负责登录验证。他仅仅收集用户名、密码、域的信息，进行一些特定的操作。当 *NT* 启动 *LOGON* 时，*LOGON* 会加载一些 *DLL*。恶意攻击者可能将带有恶意代码的 *DLL* 混入其中，使系统遭到破坏。由于攻击在用户登录时发生，且不能直接探测到，需要打开注册表对很多项进行分析，所以攻击隐蔽，危险性高。

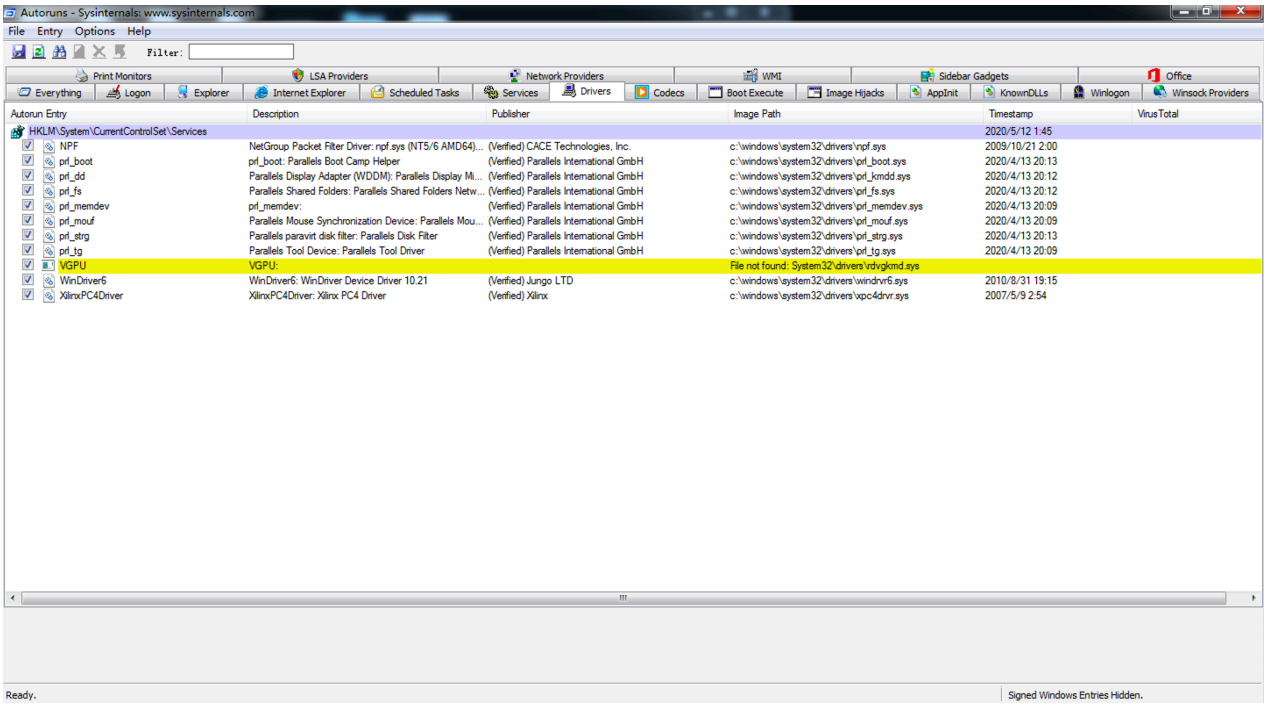
Internet Explorer：对应的是IE所有浏览器帮助对象（*BHO*）、网络 *URL* 地址搜索钩子、各类 *IE* 工具条以及 *IE* 常用工具栏按钮所对应的注册表子项和注册表值项值。

Scheduled Tasks：在 *Windows* 中，系统有一项重要的 **“计划任务”** 功能，通过设置“计划任务”，可以将每天或某一天的某个时间需要做的事拟成计划，到约定的时间，不管在电脑上进行什么工作，系统都会提醒你或者启动设定好的任务程序。通过“开始->程序->附件->系统工具”可以找到并打开“计划任务”窗口。如果攻击者在“计划任务”中插入恶意程序让其定时自动执行，将使系统遭到破坏。不过由于“计划任务”有图形界面，用户可以打开进行快速检查有无恶意程序，所以隐蔽性低。



Services：系统服务是一种应用程序类型，它在后台运行。服务应用程序通常可以在本地和通过网络为用户提供一些功能，例如客户端/服务器应用程序、Web服务器、数据库服务器以及其他基于服务器的应用程序。系统服务一般在后台运行。与用户运行的程序相比，服务不会出现程序窗口或对话框，只有在任务管理器中才能观察到它们的身影。*Service* 程序和普通的应用程序有一个根本的区别：*Service* 程序可以在无用户登录和用户已经注销的情况下运行，而应用程序在没有用户注销的时候是会被终止的。由于具备开机自启动功能，而且依靠 ROOTKIT 技术可-以隐蔽运行，所以是隐蔽性高，危险性高。

Driver：驱动程序一般指的是设备驱动程序（Device Driver），是一种可以使计算机和设备进行相互通信的特殊程序。相当于硬件的接口，操作系统只有通过这个接口，才能控制硬件设备的工作，假如某设备的驱动程序未能正确安装，便不能正常工作。然而硬件驱动程序属于操作系统内核态进程，权限高，若攻击者将恶意进程冒充硬件驱动程序，将可以造成很大危害，所以隐蔽性高而危害性大。



普通进行自启动项的查看

“启动”文件夹

“开始→程序”，会发现一个“启动”菜单，这就是最经典的Windows启动位置，右击“启动”菜单选择“打开”即可将其打开，如所示，其中的程序和快捷方式都会在系统启动时自动运行。最常见的启动位置如下：

```
当前用户：<\documents and settings\用户名\「开始」菜单\程序\启动>
所有用户：<\documents and settings\allusers\「开始」菜单\程序\启动>
```

注册表启动项

注册表是启动程序藏身之处最多的地方，主要有以下几项：

Run 键

Run键是病毒最青睐的自启动之所，该键位置是`[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]`、`[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run]`、`[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run]`，其下的所有程序在每次启动登录时都会按顺序自动执行。

RunOnce 键

`RunOnce`位于`[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce]`和`[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce]`键，与`Run`不同的是，`RunOnce`下的程序仅会被自动执行一次。

RunServicesOnce 键

`RunServicesOnce` 键位于 `[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce]` 和 `[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce]` 下，其中的程序会在系统加载时自动启动执行一次。

RunServices 键

`RunServices` 继 `RunServicesOnce` 之后启动的程序，位于注册表 `[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices]` 和 `[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices]` 键。

load键

`[HKEY_CURRENT_USER\Software\Microsoft\WindowsNT\CurrentVersion\Windows]` 下的 `load` 键值的程序也可以自启动。

Winlogon键

该键位于位于注册表 `[HKEY_CURRENT_USER\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon]` 和 `[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon]`，注意下面的 `Notify、Userinit、Shell` 键值也会有自启动的程序，而且其键值可以用逗号分隔，从而实现登录的时候启动多个程序。

其他注册表位置

还有一些其他键值，经常会有一些程序在这里自动运行，如：
`[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System\Shell]`
`[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad]`
`[HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\System\Scripts]`
`[HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\System\Scripts]`

Windows 自启动项查看软件的实现

开发环境

由于目前只有 *macOS* 机器，因此只能使用虚拟机进行开发。

物理主机系统：*macOS Catalina 10.15.4*

虚拟机系统：*Windows 10 专业版 x64*

计算机名：*691B*

虚拟机软件：*Parallels Desktop 15 for Mac Pro Edition, version 15.1.4 (47270)*

编程环境(IDE)：*Visual Studio 2019*

开发语言与架构

语言：*C++*

架构：*MFC* 应用

界面设计

由于没有使用过 `C#` 语言，而 `C` 也只在操作系统进行底层编程时使用过，所以我在`C/C++/C#`这三种可选的编程语言中选择了使用相对较多的 `C++`。但由于这学期的其他项目一直都在用 `python` 进行编程，对 `C++` 已经比较生疏，且也没有使用 `C++` 开发过有图形界面的程序，所以开发过程可谓充满艰难。

一开始是在网上查找相关资料：`如何使用 C++ 进行图形界面开发`，学着直接使用 `windows API` 进行图形界面开发。然而花了很多精力也达不到好的效果，特别是要与 Windows 自启动项查看结合。在与同学进行交流后，了解了 *Visual Studio* 自带的 *MFC* 框架。

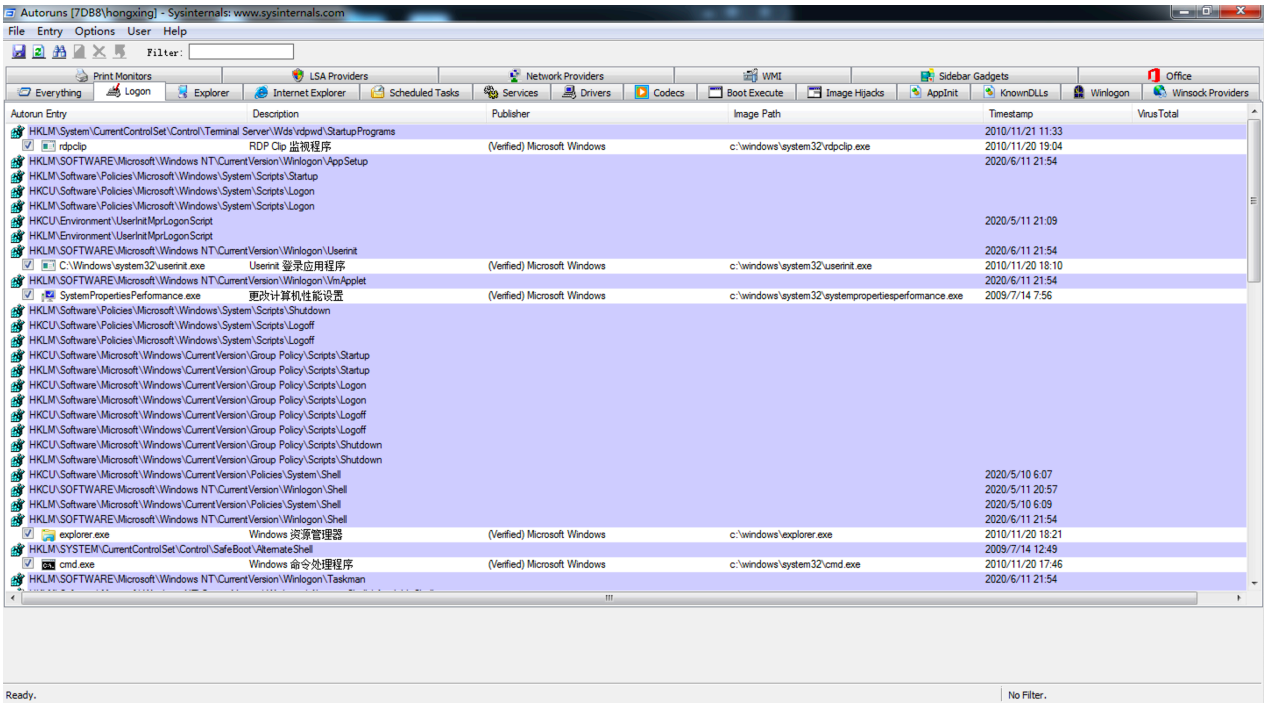
MFC (Microsoft Foundation Classes) 是微软基础类库的简称，是微软公司实现的一个 `C++` 类库，主要封装了大部分的 `windows API` 函数。*MFC* 除了是一个类库以外，还是一个框架，在 *vc++* 里新建一个 *MFC* 的工程，开发环境会自动帮你产生许多文件，同时它使用了`mfcxx.dll`。xx是版本，它封装了 `mfc` 内核，所以你在你的代码看不到原本的 `SDK` 编程中的消息循环等等东西，因为 *MFC* 框架帮你封装好了，这样你就可以专心的考虑你程序的逻辑，而不是这些每次编程都要重复的东西，但是由于是通用框架，没有最好的针对性，当然也就丧失了一些灵活性和效率。但是 *MFC* 的封装很浅，所以效率上损失不大。

由于我在程序前端开发和 `MFC` 都不熟悉，所以我前期大量的时间花在学习 `C++ MFC` 的控件与前端界面的实现。本是想用 `Autoruns` 程序作为模板，但最后能力有限未能实现，于是改为基于 `对话框` 编程。

自启动种类

Logon

一开始仅查找 `Autoruns` 软件中显示的我的 `Windows7` 虚拟机中的几个键：`HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell`、`HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`、`HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components`、`HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Components`。但和同学交流后想起老师说过的 `Autoruns` 似乎会自动隐藏空键，所以赶忙把 `Autoruns` 的设置调为空项也能显示，果然多出了许多键：

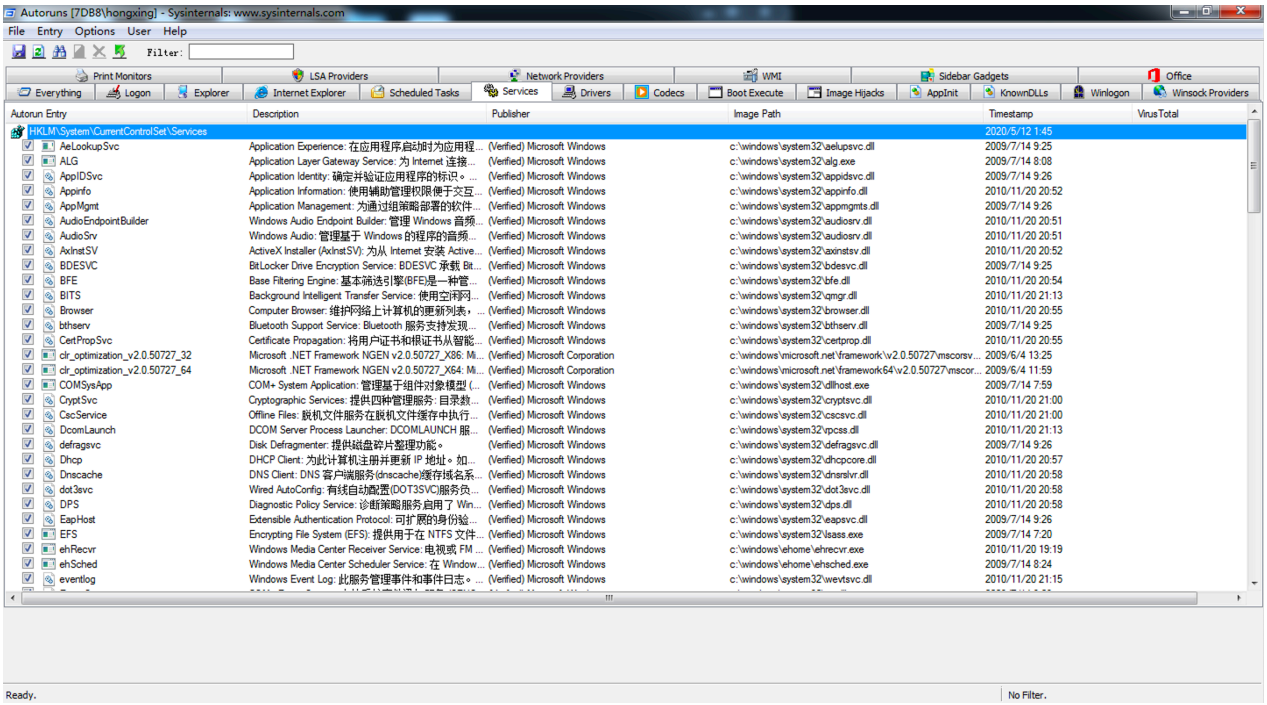



```
"HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit",
"HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\rdpwd\StartupPrograms",
"HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\AppSetup",
"HKLM\Software\Policies\Microsoft\Windows\System\Scripts\Startup",
"HKCU\Software\Policies\Microsoft\Windows\System\Scripts\Logon",
"HKLM\Software\Policies\Microsoft\Windows\System\Scripts\Logon",
"HKCU\Environment\UserInitMprLogonScript",
"HKLM\Environment\UserInitMprLogonScript",
"HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit",
"HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\VmApplet",
"HKLM\Software\Policies\Microsoft\Windows\System\Scripts\Shutdown",
"HKCU\Software\Policies\Microsoft\Windows\System\Scripts\Logoff",
"HKLM\Software\Policies\Microsoft\Windows\System\Scripts\Logoff",
"HKCU\Software\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Startup",
"HKLM\Software\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Startup",
"HKCU\Software\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Logon",
"HKLM\Software\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Logon",
"HKCU\Software\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Logoff",
"HKLM\Software\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Logoff",
"HKCU\Software\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Shutdown",
"HKLM\Software\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Shutdown",
"HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\Shell",
```

```
"HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell",
"HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Shell",
"HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell",
"HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell",
"HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Taskman",
"HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\AlternateShells\AvailableShells",
"HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\Software\Microsoft\Windows\CurrentVersion\Runonce",
"HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\Software\Microsoft\Windows\CurrentVersion\RunonceEx",
"HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\Software\Microsoft\Windows\CurrentVersion\Run",
"HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp\InitialProgram",
"HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run",
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run",
"HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run",
"HKCU\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run",
"HKCU\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnceEx",
"HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce",
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnce",
"HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce",
"HKCU\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnce",
"HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load",
"HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run",
"HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run",
"HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run",
"HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components",
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Components",
"HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\IconServiceLib",
"HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\Software\Microsoft\Windows\CurrentVersion\Runonce",
"HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\Software\Microsoft\Windows\CurrentVersion\RunonceEx",
"HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\Software\Microsoft\Windows\CurrentVersion\Run",
"HKLM\SOFTWARE\Microsoft\Windows CE Services\AutoStartOnConnect",
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows CE Services\AutoStartOnConnect",
"HKLM\SOFTWARE\Microsoft\Windows CE Services\AutoStartOnDisconnect",
"HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows CE Services\AutoStartOnDisconnect",
```

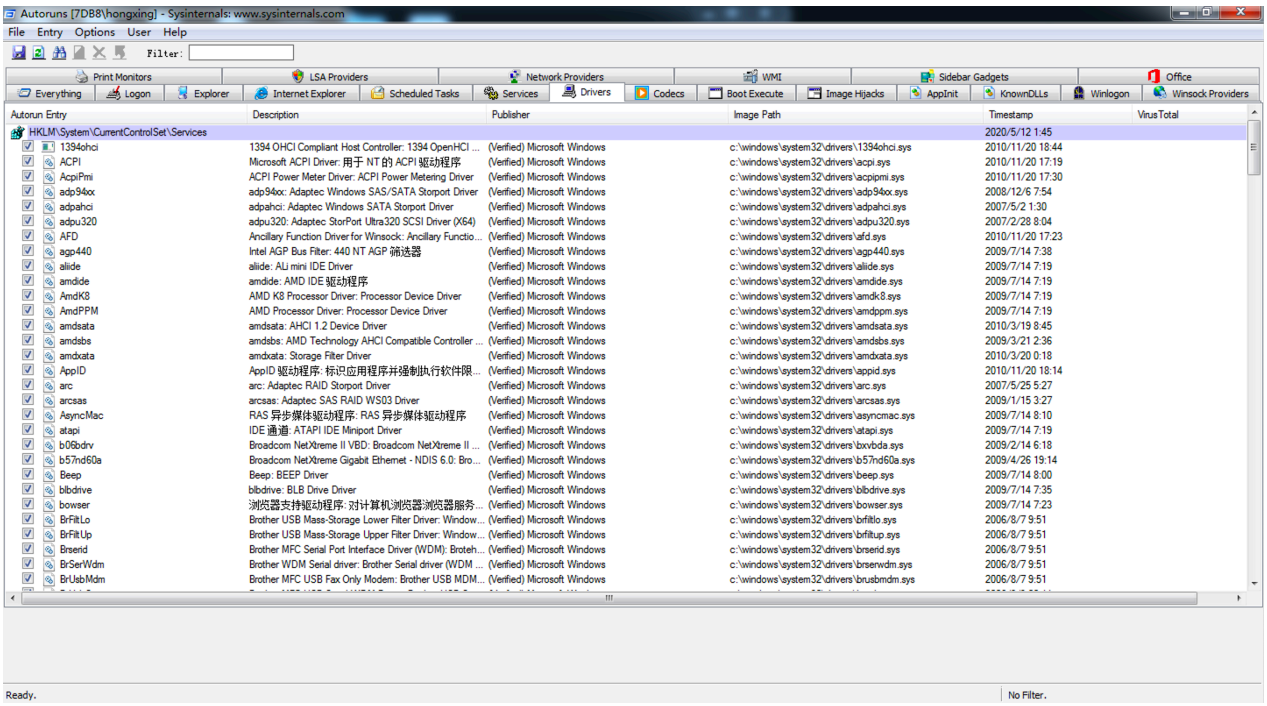
Services

查看 `Autoruns` 的 `Services` 项发现通过Services系统服务实现自启动的条目全部在注册表的 `HKLM\System\CurrentControlSet\Services` 这一键下。



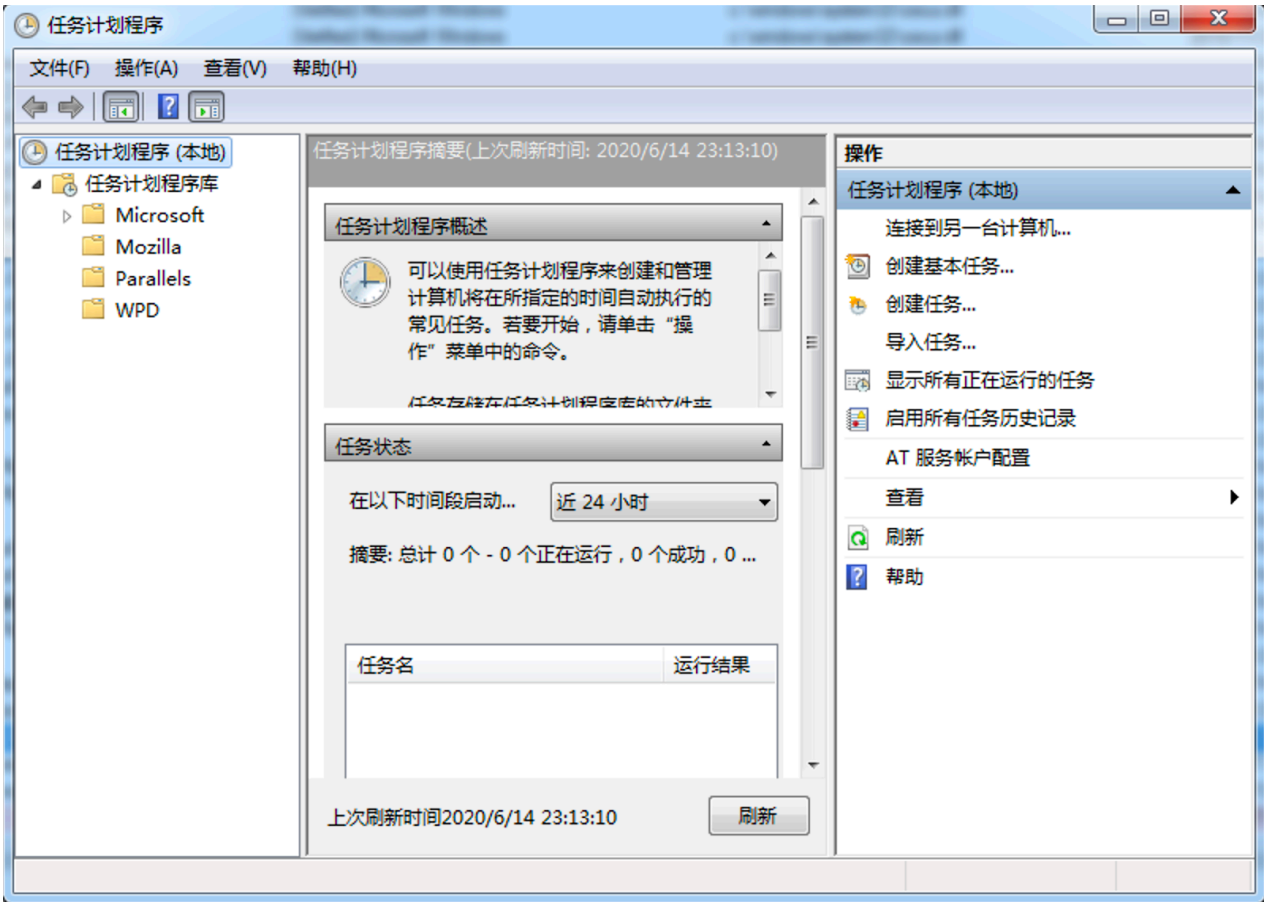
Drivers

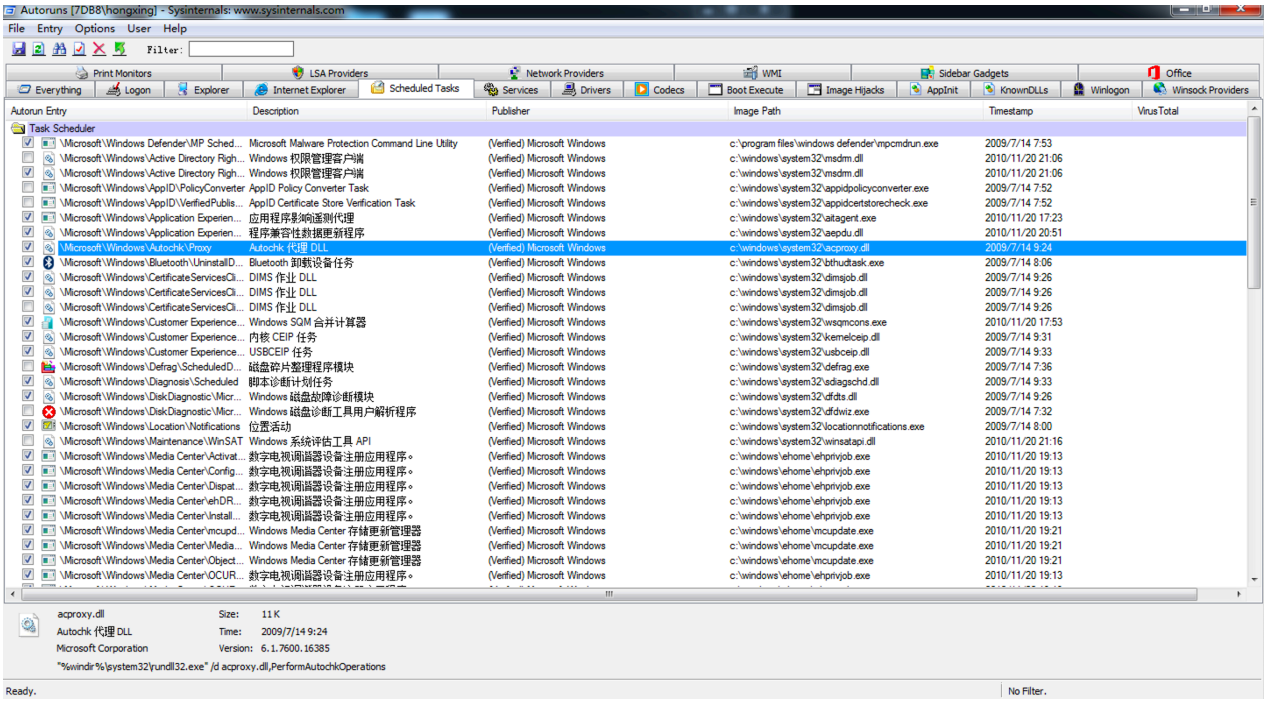
系统驱动程序和Services系统服务在同一个键下，所以只按照可执行文件名是否是以 `.sys` 结尾就可以区分是 `Services` 还是 `Drivers`。



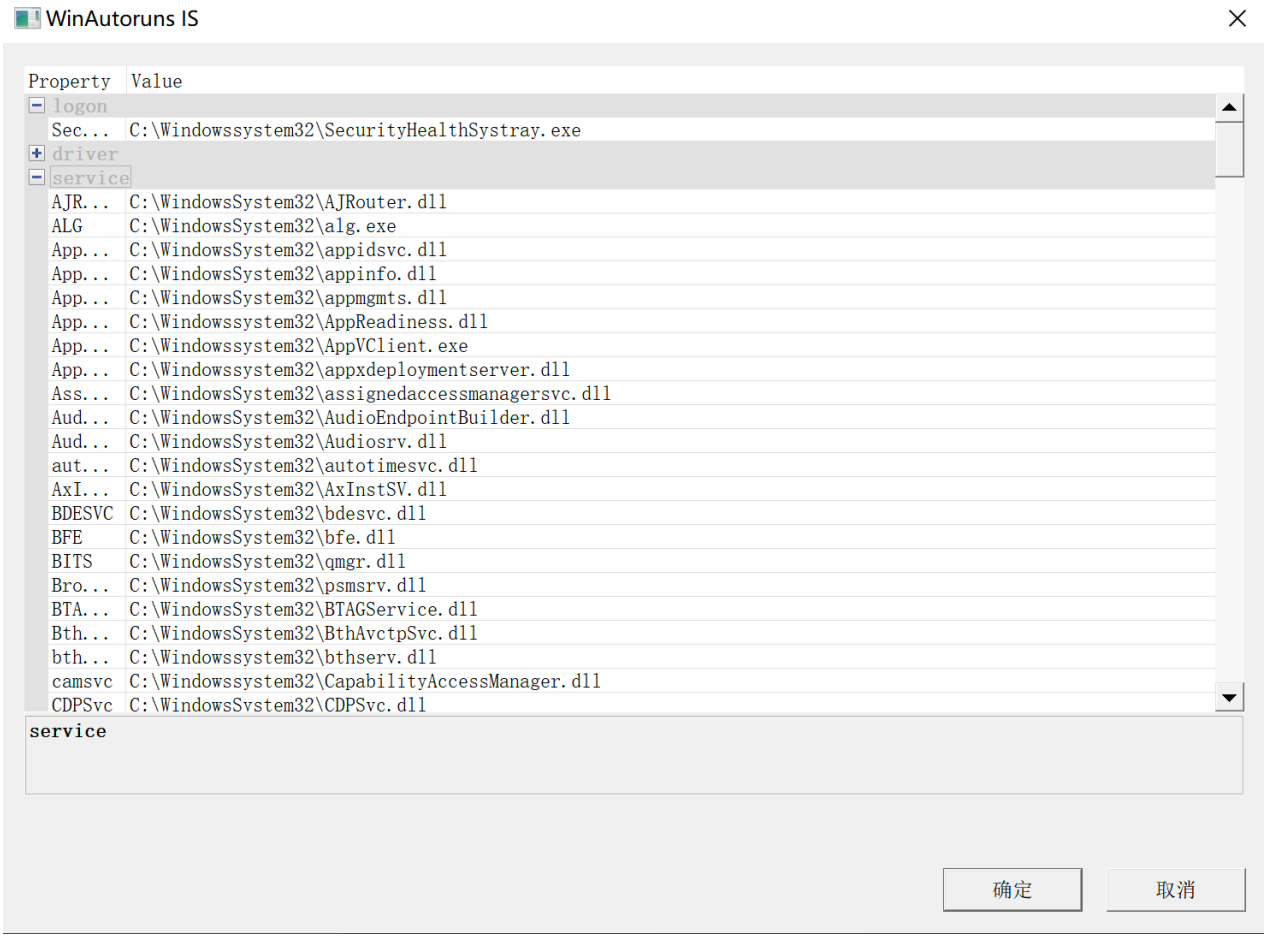
Scheduled Tasks

打开“任务计划”界面可以看到本地任务计划程序库，上网查询得知其位置是`C:\Windows\System32\Tasks`，读取该位置查询即可。





本来还想实现其他的自启动项查看，但无奈时间不够、能力有限，未能成行。



问题与感想

本门课程的大作业中遇到的最大问题就是对于 *C/C++/C#* 编程不熟悉，一直用惯了 *python* 方便的编程，对于这类较低层的语言编程便显得力不从心，指针、结构体、类型转换以及 *Visual Studio 2019* 的 *C++* 版本等问题弄得我头秃。不过幸好任务本身不难，还有同学进行交流，了解使用 *MFC* 框架后也能较顺利地完成任务，只是最后的页面还是不尽如人意。另外由于一直使用 *macOS* 系统，对 *Windows* 系统已经不是很熟悉，*Visual Studio 2019* 的使用也不熟练，弄崩了好多次。

这个大作业让我有机会深入 *Windows* 的底层，去查看系统中的自启动项，来发现其中的“不法分子”，实际体验了系统安全人员的工作。另外还有机会使用 `SysinternalsSuite` 工具集进行了不少实验，有趣的同时也对 `Windows` 安全的了解更为深入。另外锻炼了 `C++` 编程的能力，学习了微软的 `MFC` 框架。