**恶意代码分析与防治技术实验报告**

**Lab3**

1. **实验环境**

Windows7，VMWARE，Windows11

1. **实验工具**

STRINGS, IDAPro,PEVIEW，EXEInfo，YARA

1. **实验内容**

Lab 3-1

1. What are this malware’s imports and strings?

该恶意代码是被加过壳的，只有一个导入函数——ExitProcess，字符串大多都是明文，其中

比较可疑的有注册表位置、域名、admin（管理员）、VideoDriver、WinWMX32、vmx32to64.exe

（检测虚拟机）等。

分析过程：

使用查壳工具 PEid 进行查壳，发现该程序被加壳，且使用脱壳工具 LinxerUnpacker 脱壳失败。

使用 Dependence Walker 分析，可以看到只有一个导入函数——ExitProcess。

使用 Ida Pro 查看字符串，如下所示：

其中比较可疑的有注册表位置、域名、admin（管理员）、VideoDriver、WinWMX32、

vmx32to64.exe（检测虚拟机）等。

2. What are the malware’s host-based indicators?

该恶意代码创建了一个名为 WinVMX32 的互斥量，并且在 C:\Windows\System32 的目录下

创建了一个自己的副本：vmx32to64.exe，同时向注册表中写入了一个新路径用于在系统启

动时自动运行 vmx32to64.exe。

分析过程：

下面我们将开始动态分析技术。

首先运行 Process Monitor 工具，并清除所有事件；启动 Process Explorer；使用 ApateDNS 和

Netcat 配置出一个虚拟网络，监听 80 和 443 端口，以及用于抓包的 Wireshark。

接下来开始运行恶意代码 Lab03-01，在 Process Explorer 中检查进程。选择 Lab03-01.exe 进

程项目，查看它的 Handles 视图，可以发现恶意代码已经创建了一个名为 WinVMX32 的互斥

量。

查看 procmon，通过设置过滤信息：

来查看恶意代码对文件系统和注册表的修改，得到如下十条结果：

其中有一个 WriteFile 操作的条目，该操作在 C:\WINDOWS\system32 目录下创建了一个程序，

对比发现，Lab03-01.exe 与这个新创建的 vmx32to64.exe 的文件大小、Hash 值都是一样的，

这说明 Lab03-01.exe 是在这个目录下创建了自己的一个副本。

其余九个是关于 RegSetValue 操作的条目，细心观察可以发现，其中有八条都是在设置种子，

这是典型的噪声。而图中的第三条操作是比较不一样的，双击以后可以得到如下信息：

此时可以发现在 C 盘的 system32 目录下出现了一个名为 vmx32to64.exe 的程序，同时，恶

意代码向注册表中写入了一个路径：

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\VideoDriver:C:\WINDOWS\system32\vmx32to64.exe 该新建注册表项用于在系统启动时自动运行 vmx32to64.exe。

3. Are there any useful network-based signatures for this malware? If so, what are they?

该恶意代码会对 www.practicalmalwareanalysis.com 这个域名进行解析和访问，且会在 443

端口广播一些看上去是随机的、每次都不相同的数据。

分析过程：

下面我们来查看网络分析工具。

首先检查 ApateDNS，可以看到恶意代码对 www.practicalmalwareanalysis.com 域名的请求（与

之前查找到的字符串中的域名相匹配）。

再检查 Netcat 的结果，发现 80 端口并没有什么反馈，而在 443 端口监听到了一些信号，且

多次运行得到的数据是不同的，可见是随机产生的（推测与之前注册表中设置的不同种子有

关）。

（使用 Kali 监测的）

Lab 3-2

1. How can you get this malware to install itself?

Windows 系统中包含一个用于运行 DLL 的 rundll32.exe 程序，在该 dll 目录下运行命令

rundll32.exe Lab03-02.dll , installA，即可运行恶意代码导出的 installA 函数，将该恶意代码安

装成一个服务。

分析过程：

首先使用 PEView 进行静态分析，看到该 DLL 有五个导出函数，其中 ServiceMain 说明该恶意

代码需要被手动安装成一个服务。同时，这个 dll 文件也提供了 Install 和 installA 两个进行安

装的函数，即在使用 rundll32.exe 时用到的函数。

该恶意代码有很多导入函数，其中包含了一些服务操作函数（如 CreateService）、注册表操

作函数（如 RegSetValueEx）、网络操作函数（如 HeepSendRequest）等，表明该恶意代码可

能需要安装服务、修改注册表、使用网络等等。

使用 Strings 工具检查字符串列表，由于列表很长，只截取值得关注的部分，可以看到，除

了上面提到的 Exports 和 Imports 之外，还出现了注册表位置、域名、IPRIP、serve.html、sleep、

GetSystemTime 等字符串。

通过以上分析，我们推测这个恶意代码需要使用导出函数 installA 将自身安装成为一个服务。

下面开始基本的动态分析，在安装恶意代码之前，我们先打开 Regshot 工具，以监视注册表

的变化，并打开 Process Explorer 来监控在系统上运行的进程。

然后我们使用如下命令来安装恶意代码：

可以看到此时 rundll32.exe 正在运行。

等到 rundll32.exe 终止运行时，用 Regshot 工具做第二个快照进行对比，得到以下结果：

其中，Keys Added 中 HKLM\SYSTEM\CurrentControlSet\Services\IPRIP 表示恶意代码将自身安

装为 IPRIP 服务，并依赖一个可执行文件来执行它；Values Added 中

HKLM\SYSTEM\ControlSet001\Services\IPRIP\ImagePath: "%SystemRoot%\System32\svchost.exe

-k netsvcs"将 ImagePath 设置成 svchost.exe，意味着这个恶意代码将会在 svchost.exe 进程中

被调用。

至此，该恶意代码已经被安装为一个服务。

2. How would you get this malware to run after installation?

使用命令 net start IPRIP 可以运行该恶意代码。

分析过程：

在运行恶意代码之前，首先运行 Process Monitor 工具，并清除所有事件；启动 Process

Explorer；使用 ApateDNS 和 Netcat 配置出一个虚拟网络，并监听 80 端口。

执行命令 net start IPRIP 运行恶意代码：

3. How can you find the process under which this malware is running?

使用进程浏览器 Process Explorer 即可知道当前有哪些进程在运行该服务，通过其中的 Find

工具，可以查看到，只有 PID 为 1048 的 svchost.exe 进程在运行该服务，与前面的分析一致。

4. Which filters could you set in order to use procmon to glean information?

在 procmon 中，可以设置 Process Name 来过滤，但在上面对 Process Explorer 的分析过程中

发现，有两个名为 svchost.exe 的程序，重名了，于是这里采用在 Process Explorer 中获得的

PID（可以唯一标识一个程序）来过滤。

5. What are the malware’s host-based indicators?

该恶意代码将自身安装为 IPRIP 服务，并且将自己写入注册表中。

6. Are there any useful network-based signatures for this malware?

该恶意代码会对 www.practicalmalwareanalysis.com 这个域名进行解析和访问，且会在 80

端口向 serve.html 执行 GET 请求。

分析过程：

下面我们来分析网络行为。

检查 ApateDNS，我们发现该恶意代码向 practicalmalwareanalysis.com 执行了一次 DNS 请求；

观察 Netcat 结果，发现恶意代码在 80 端口执行了一次 GET 请求，且多次测试结果一致。分

析这个结果，可以得知，恶意代码的请求对象是 serve.html，请求方式是 GET，User-Agent

为自己的用户名+Windows XP 6.11。

值得注意的是，以上网络通信行为在启动服务后 60 秒才看到，推测这与之前在检查字符串

时看到的 sleep 函数有关。

Lab 3-3

1. What do you notice when monitoring this malware with Process Explorer?

该恶意代码在创建子进程 svchost.exe 后退出，留下 svchost.exe 继续运行，达到了替换

的目的。

分析过程：

我们先对该恶意代码进行静态分析，通过 strings 工具查看字符串，如下：

得到的都是乱码及大量的重复字符串，于是我们推测该程序被混淆了，于是开始动态分析。

在运行恶意代码之前，首先运行 Process Monitor 工具，并清除所有事件；启动 Process

Explorer。

执行恶意代码，观察 Process Explorer 发现，Lab03-03.exe 刚开始运行几秒钟便退出了，并创

建了一个子进程 svchost.exe。

过滤 procmon 的结果，分析 Lab03-03.exe 的行为，发现它可以创建进程、加载映像、创建

文件、关闭文件等，通过进一步对 operation 的过滤，我们可以发现它确实创建了一个子进

程 svchost.exe，PID 为 3060。

接下来我们去查看 svchost.exe 的相关信息，发现该文件源程序大小仅有 14KB，然而从上面

Process Explorer 中得到的结果来看，该文件实际大小是比 14KB 大得多的。

分别查看 image 和 memory 中的 strings，差别很大，且在 memory 中有一个.log 文件，还有

[SHIFT]、[ENTER]等，由此推测这个程序应该有一个日志记录，且与键盘操作有关。同时因

为内存和磁盘中的 strings 区别很大，还可以推测出这个程序应该是做了进程替换。

找到该日志文件（与 Lab03-03.exe 在同一目录下），是在记录键盘的操作。

2. Can you identify any live memory modifications?

通过上面的分析对比，我们知道，svchost.exe 在 Image 和 Memory 上是非常不同的，Memory

拥有 practicalmalwareanalysis.log、[SHIFT]、[ENTER]等字符，而 Image 中没有。

3. What are the malware’s host-based indicators?

该恶意软件在 windows\system32 目录下创建了 svchost.exe，且在 Lab03-03.exe 的同一目录

下创建了一个 practicalmalwareanalysis.log 日志文件。

4. What is the purpose of this program?

该恶意软件对进程 svchost.exe 进行了替换，来实现一个击键记录器。

Lab 3-4

1. What happens when you run this file?

双击运行这个恶意代码的时候，它会打开一个命令行窗口，迅速关掉，并进行自我删除。

分析过程：

我们先对该恶意代码进行静态分析，通过 strings 工具查看字符串，如下，推测应该没有加

壳。

值得注意的有：SunMon...Sat、JanFeb...Dec 等日期、域名、注册表位置、CMD/GET/SLEEP 等

操作、DOWNLOAD/UPLOAD 等命令字符串、HTTP/1.0，以及-cc、-re、-in 等命令行参数等，

推测可能是在命令行中运行某些指令，来修改注册表、联网等。

使用 PEiD 工具查看，证实了确实没有加壳。

下面我们将开始动态分析技术。

首先运行 Process Monitor 工具，并清除所有事件；启动 Process Explorer；并配置出一个虚

拟网络。

双击运行这个恶意代码，发现它打开了一个命令行窗口，然后窗口很快就关闭了，且在

Process Explorer 中监测不到该程序了，且在源目录与回收站中都无法找到该程序，应该是进

行了自我删除。

过滤 procmon 的结果，分析 Lab03-04.exe 的行为，发现他多次修改注册表。

使用 RegShot 工具查看其对注册表的修改，结果如下：

可以看出该恶意代码增加了注册表的表项，其中比较值得注意的是：其增加的位置是

ShellNoRoam\MUICache 加上程序的绝对路径，查阅资料得知这个键值的作用是：

1、防火墙阻止程序名称与实际程序描述不一致时，只能通过注册表更改数值数据信息（如

绝对路径不变，即不会建立新的键值时），也可直接删除整个键，系统会重新建立新的键值。

2、个性化系统 exe 名称显示，更改数值数据信息。 如：查找到“回收站”键值更改为“垃圾

桶”等。其它程序也可做类似的更改。

2. What is causing the roadblock in dynamic analysis?

结合在分析字符串时的发现，我们推测可能不能用双击的方式直接运行，而应该用命令行

运行，并提供一些必要的参数等。

3. Are there other ways to run this program?

我们尝试使用命令行运行该程序，并加上一些命令行参数，结果都是该恶意代码进行自我

删除。

分析过程：

通过以上分析，我们决定使用命令行的方式运行该程序，如下：

但仍然会进行自我删除，于是决定再加上 strings 查找到的-cc、-re、-in 等命令行参数，还是

同样的结果。

修改 procmon 的过滤条件，找到了该程序自我删除的相关操作。

至此，使用基础的动态分析方法无法再有效的推进下去。

1. **实验心得**

**本次实验，我熟悉了对恶意代码分析工具有了更深入的理解。同时得到了充分的锻炼，并开始对本门课程的实验开始得心应手，做实验的速度越来越快了。**