**恶意代码分析与防治课程实验报告**

**实验14**

专 业 信息安全

学 号 2113662

姓 名 张丛

班 级 信安一班

1. **实验目的**

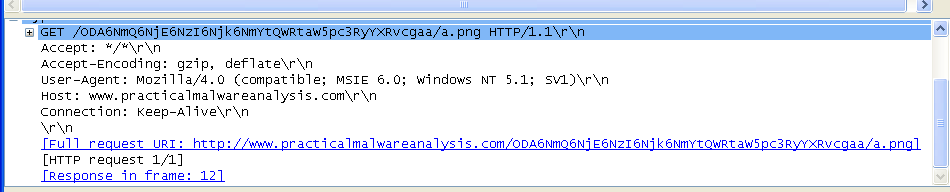
完成课本Lab 14的实验内容。

1. **实验过程**

##### Lab 14-01

### 1.恶意代码使用了哪些网络库？它们的优势是什么？

使用WireShark进行动态分析。

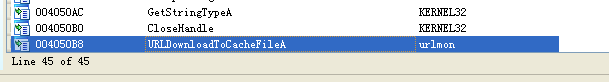


使用另外的机器进行分析对比可知，User-Agent不是硬编码。

请求的URL值得注意。

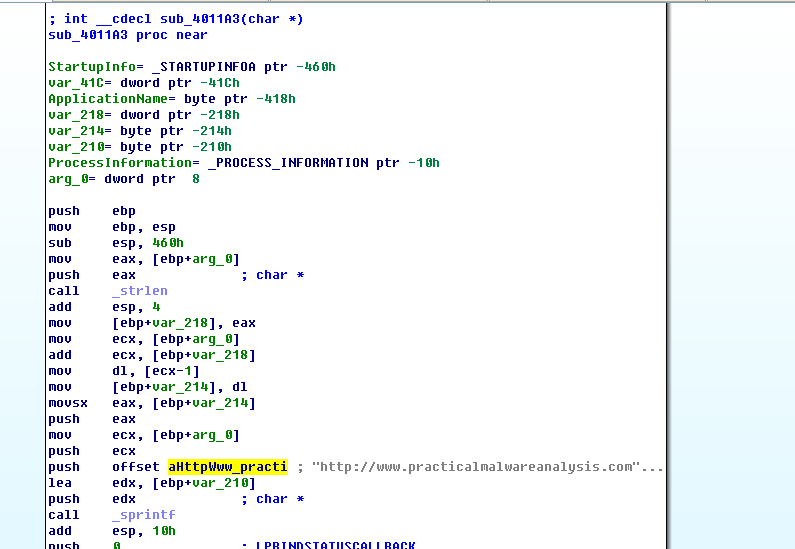
回答：使用了URLDownloadToCacheFileA函数，该函数使用了COM接口，当恶意代码使用COM接口时，HTTP请求中的大部分内容都来自Windows内部，无法有效地使用网络特征进行针对性的检测。

### 2.用于构建网络信令的信息源元素是什么，什么样的条件会引起信令的改变？



可以通过该函数去寻找是什么构造了URL。

通过交叉引用，我们发现了这样的一个函数。



通过分析可知它的作用是从一个网址下载文件，然后运行它，我们将该函数命名为downandrun。

请求网址有两部分是需要填充的。

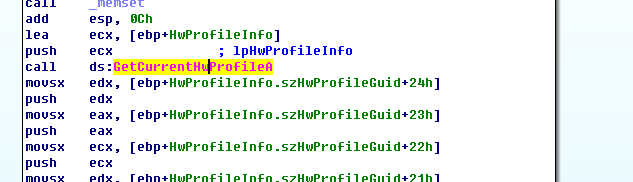
:::info

http://www.practicalmalwareanalysis.com/%s/%c.png

:::

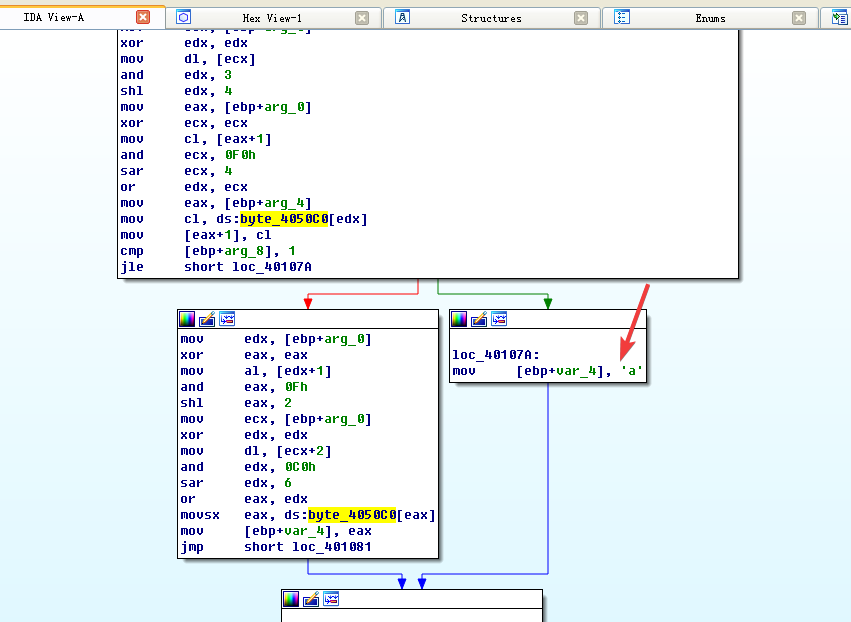
%s是参数，%c是这个参数的最后一个字符。

接着，我们看看是什么引用了这个函数。



这两个API函数的返回值构成了downandrun函数的参数。

几番曲折，找到了加入字符a的函数。



对加密的字符串进行Base64解密。

:::info

ODA6NmQ6NjE6NzI6Njk6NmYtQWRtaW5pc3RyYXRvcga

:::



回答：发送的信息中包含了主机的HWID和用户名；其中HWID是固定的，用户名则是可变的。

### 3.为甚攻击者可能对嵌入在网络信令中的信息感兴趣？

回答：攻击者想获取特定的用户和主机，在下一步进行针对性地攻击。

### 4.恶意代码是否使用了标准的Base64编码？如果不是，编码是如何不寻常的？

回答：不是常用的Base64编码，因为常见的base64编码在长度不够时采用“=”进行末尾填充，而此恶意程序采用‘a’进行末尾填充。

### 5.恶意代码的主要目的是什么？

回答：将被感染主机的HWID和用户名发送到到C&C，然后从网站下载并运行其他恶意代码

### 6.使用网络特征可能有效探测到恶意代码通信中的什么元素？

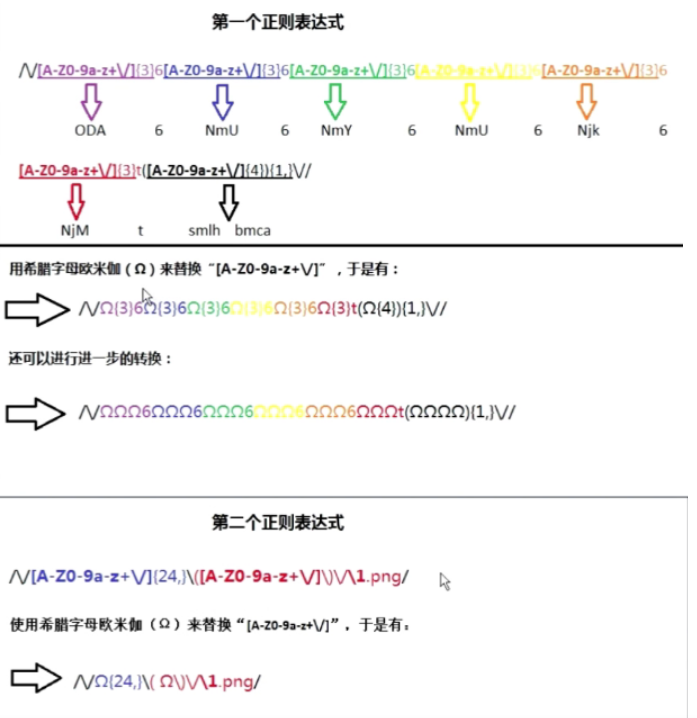
回答：域名特征、base64编码特征和HTTP请求中的URI的后缀是PNG文件，

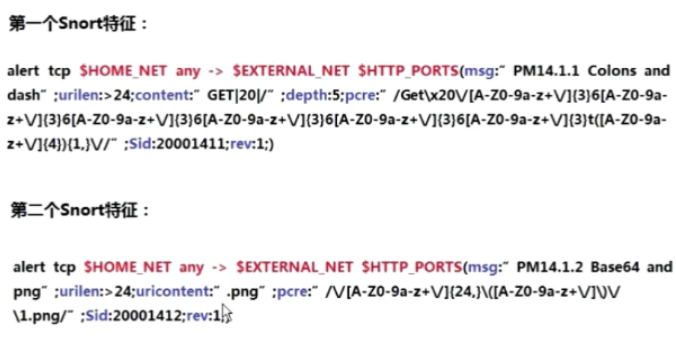
### 7.分析者尝试为这个恶意代码开发一个特征时，可能会犯什么错误？

回答：如果恶意代码分析人员没有意识到下载网址与用户操作系统有关，可能会把重点放在URL上面。文件名可能是a.png，但当用户名长度和计算机属性是三的倍数时，文件名会变得无法预测。

### 8.哪些特征集可能检测到这个恶意代码（以及新的变种）？

参考了网上的解答。



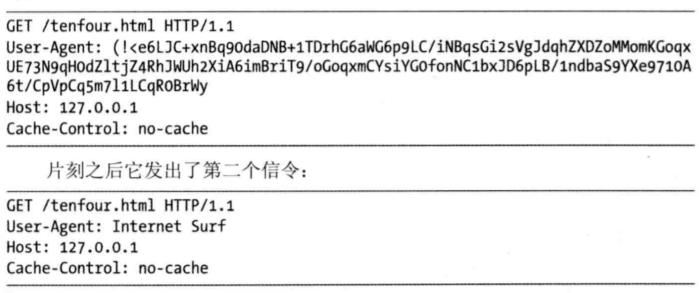


##### Lab14-02

### 1.恶意代码编写时直接使用IP地址的好处和坏处各是什么？

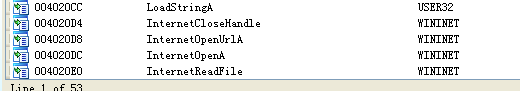
攻击者可能会发现静态IP地址比域名更难管理。使用DNS允许攻击者将他的系统部署到任意一台计算机上，仅仅改变DNS地址就尅动态重定向他的僵尸主机。对于这两种类型的基础设施，防御者有不同选项比域名更难处理。这个事实会让攻击者选择静态IP地址，而不是域名。

这里我们直接使用本书作者捕获到的信令。



### 2.恶意代码使用了哪些网络库？使用这些库的好处和坏处是什么？

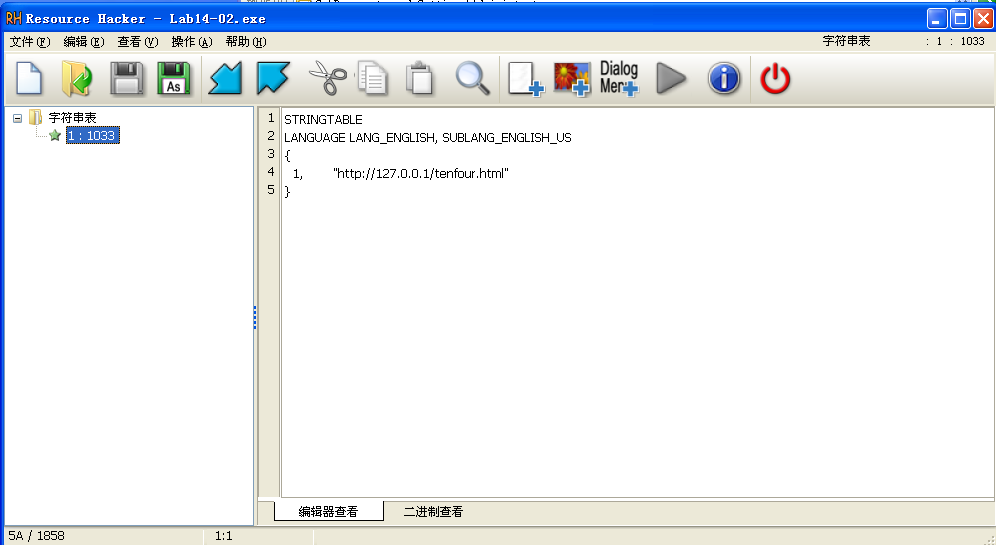
使用IDA查看。



回答：使用了WinNet库。优点是与Winsock API相比，操作系统可以提供较多的网络字段元素；缺点就是该库的网络函数调用需要提供一个硬编码的UserAgent字段。

### 3.恶意代码信令中URL的信息源是什么？这个信息源提供了哪些优势？

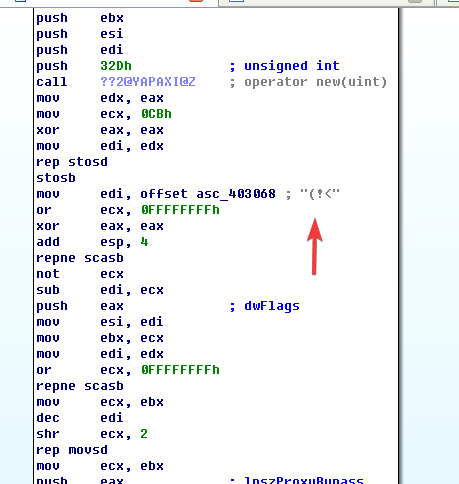
使用ResourceHacker查看资源节。



回答：PE文件中的字符串资源节包含一个用于命令和控制的URL。在不重新编译恶意代码的情况下可以让攻击者使用资源节来部署多个后门程序到多个命令与控制服务器位置。

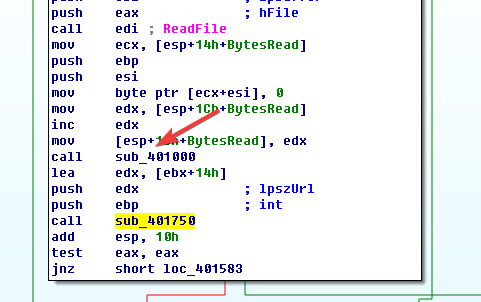
### 4.恶意代码利用了HTTP协议的哪个方面，来完成它的目的？

在函数sub\_401750中，我们看到了第一个信令中的一些信息。

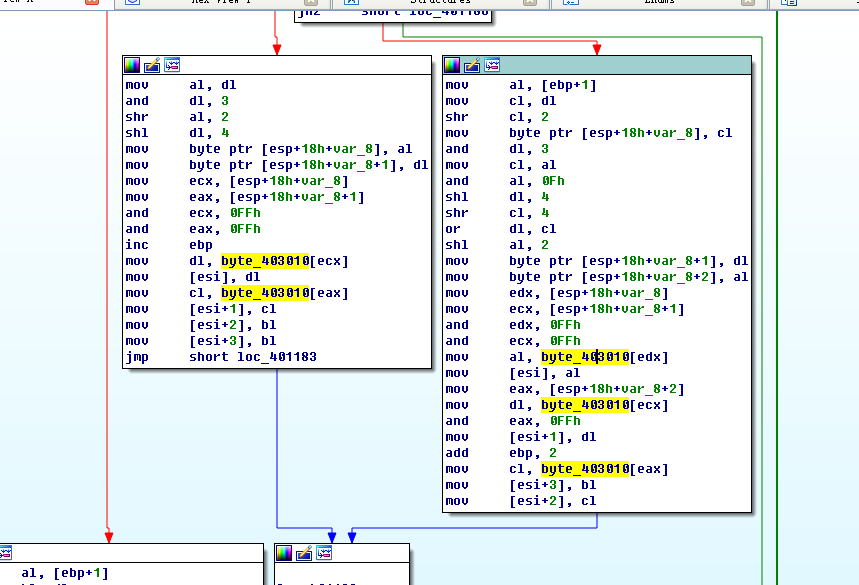


说明使用了UserAgent来发送信息，并且信息经过了加密。

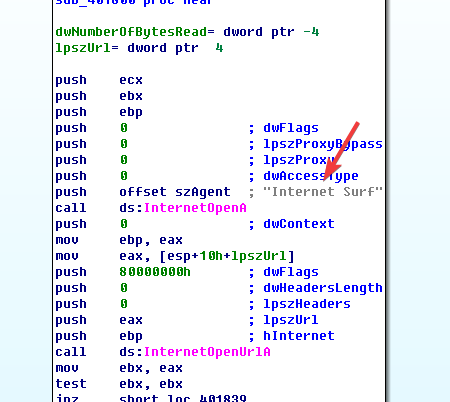
对该函数进行交叉引用。



检查该函数执行前的函数。



该函数是变种的Base64加密。



第二条信令所在位置。

回答：利用UserAgent字段来发送信息，并且该信息被加密；通过两个管道通信来实现远程控制和任意命令执行。

### 5.在恶意代码的初始信令中传输的是哪种信息？

IMG_256

初始的信令是 一个编码后的shell命令行提示。

### 6.这个恶意代码通信信道的设计存在什么缺点？

尽管攻击者对传出信息进行编码，但他并不对传入命令进行编码。此外，由于服务器必须通过User-Agent域的静态元素，来区分通信信道的两端。所以服务器的依赖关系十分明显，可以将它作为特征生成的目标元素。

### 7.恶意代码的编码方案是标准的吗？

自定义的base64编码。

### 8.通信是如何被终止的？

使用关键字exit来终止通信，退出时恶意代码会试图删除自己。

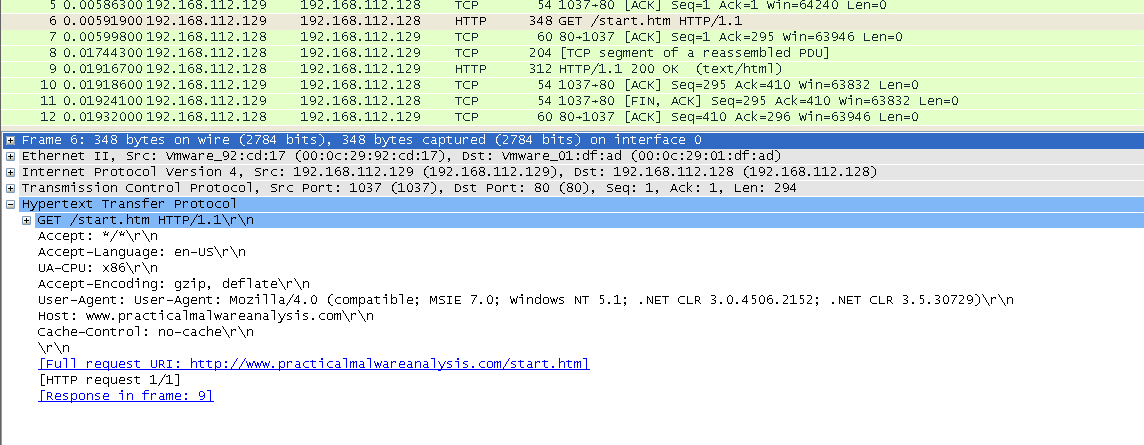
### 9.这个恶意代码的目的是什么？在攻击者的工具中，它可能会起什么作用？

这个恶意代码是一个小且简单的后门程序。它的唯一目的是给远端的攻击者提供一个shell命令接口，而通过查看出站shell命令活动的常见网络特征不能监测到它。基于它尝试删除自己这个事实，我们推断这个特殊的恶意代码可能是攻击者工具包中的一个一次性组件。

##### Lab14-03

### 1.在初始信令中硬编码元素是什么？什么元素能够用于创建一个好的网络特征？

首先使用WireShark进行动态分析。这里的User-Agent出现了重复。



再使用IDA进行静态分析。

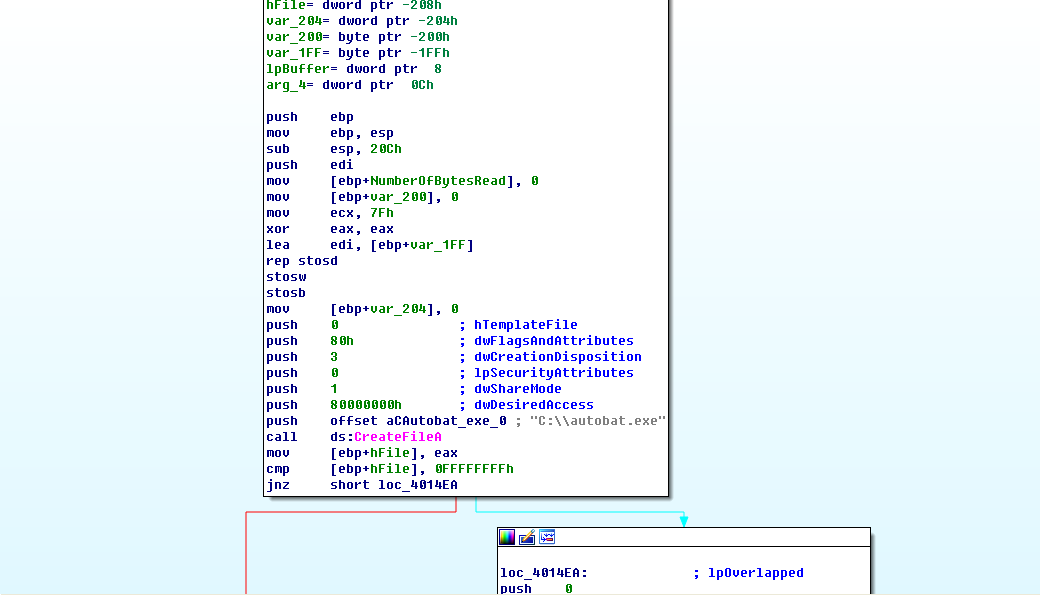


在InternetOpenUrl函数调用的上方可以看到网络信令中看到的字符串。InternetOpenUrl调用后会包含头部的标题。

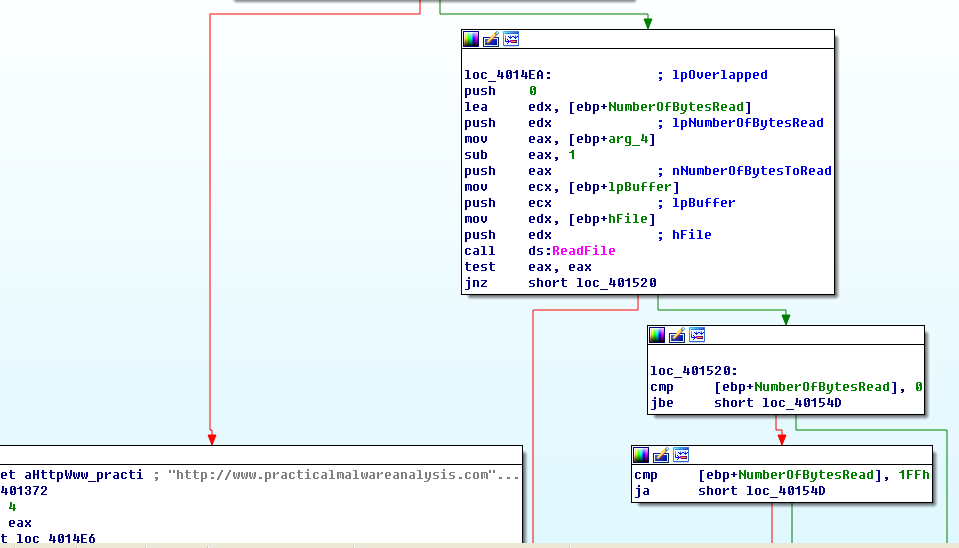
硬编码的头部包括Accept、Accept-Language、UA-CPU、Accept-Encoding和User-Agent恶意代码错误地添加了一个额外的User-Agent，在实际的User-Agent中，会导致重复字符串：User-Agent：User-Agent：Mozilla…针对完整的User-Agent头部（包括重复字符串），可以构造一个有效的检测特征。

### 2.初始信令中的什么元素可能不利于可持久使用的网络特征？

首先，找确定信令内容的函数。

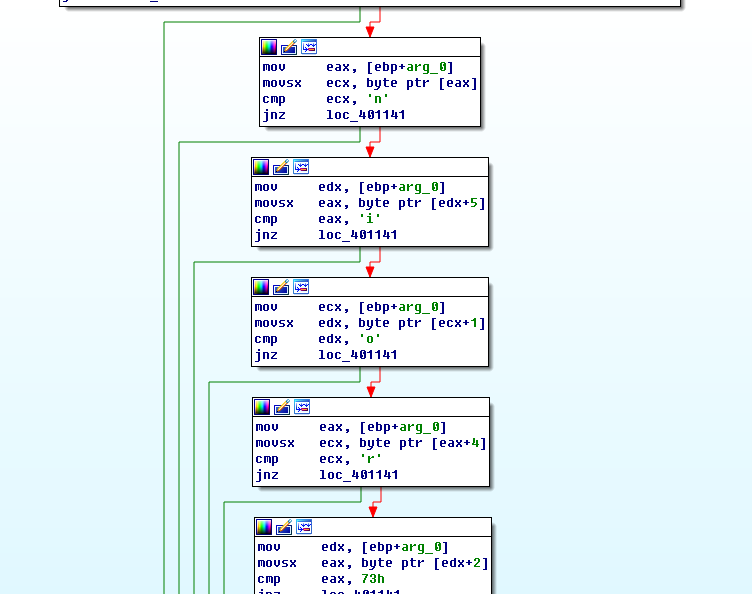


如果文件不存在，那么就返回这个信令的URL，autobat.exe是存储URL明文的配置文件。



恶意代码从Web页面上noscript标签中的某些特定组件来获得命令，这与本章中提到的注释域例子类似。使用这种技术，恶意代码可以向一个合法的网页发出信令，并且接收合法内容，使防御者区分恶意流量与合法流量变得更加困难。

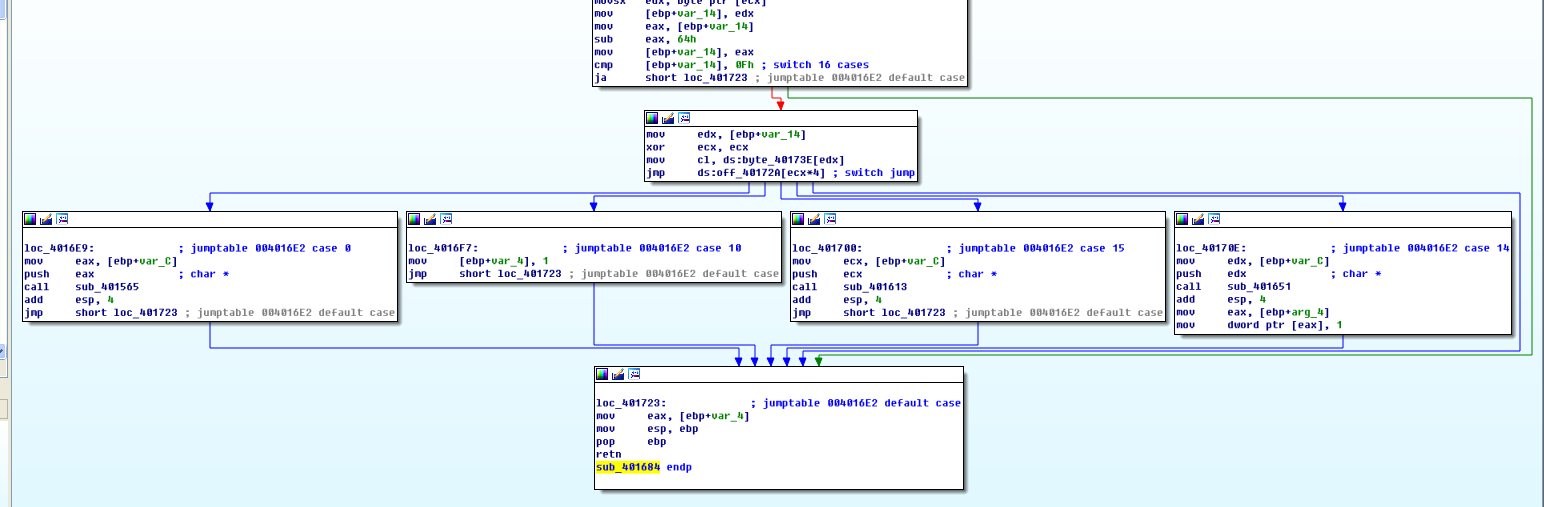
### 3.恶意代码是如何获得命令的？本章中的什么例子用来类似的方法？这种技术的有点是什么？



上图展示了使用较小连接块的输入缓冲区测试。攻击者通过将字符串比较拆分成多个较短的字符串测试，来消除明显的字符串比较，从而尝试隐藏他们正在查找的自妇产。此外，请注意请求的字符串（）被混淆了，从而避免产生明显的模式。图中是0位置上的n、5位置上的i，以及1位置上的o。

接下里面的代码是从被截断URL尾部开始对静态字符串96‘进行搜索。

然后我们分析switch函数。



‘s’函数是进行休眠。‘d’函数调用了URLDownloadToCacheFile和CreateProcess。

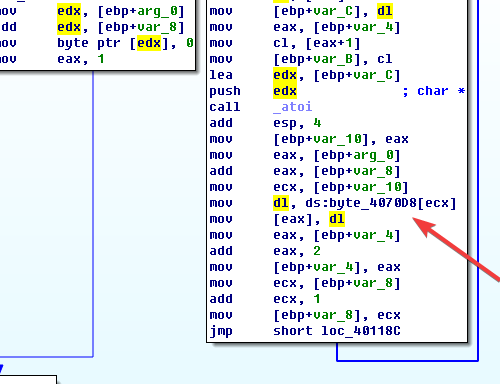
恶意代码从Web页面上noscript标签中的某些特定组件来获得命令，这与本章中提到的注释域例子类似。使用这种技术，恶意代码可以向一个合法的网页发出信令，并且接收合法内容，这使防御者区分恶意流量与合法流量变得更加困难。

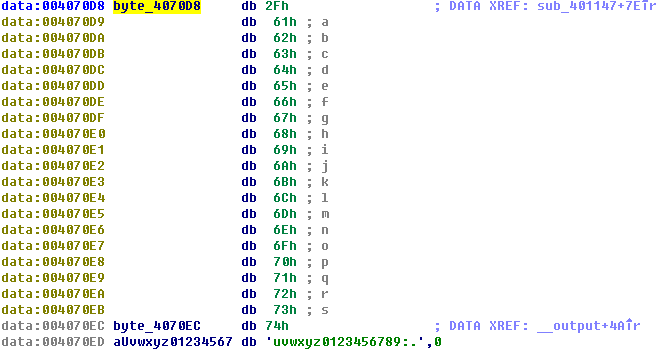
#### 4.当恶意代码接受到输入时，在输入上执行什么检查可以决定它是否是一个有用的命令？攻击者如何隐藏恶意代码正在寻找的命令列表？

要将内容解析为命令，必须包含被完整URL（包括http:/）跟随的初始noscript标签，此URL包含的域名与原始网页请求的域名相同。此URL路径必须以96结尾。域名和96（其中被截断）之间的两部分组成了命令和参数（如/command/1213141516类似的形式）。命令的第一个字母必须与提供命令相对应，在合适的时候，参数必须翻译成给定的命令中有意义的参数。

恶意代码编写者限制了可以提供有关恶意代码功能线索的字符串列表。当搜索noscript标签时，恶意代码搜索了<no，接着用独立不规则的字符比较操作来确定noscript标签。恶意代码也复用了域名所使用的缓冲区，来检查命令的内容。此外对96的字符串搜索只有三个字符，另外唯的单字符串搜索是字符/。当匹配命令时，仅仅考虑第一个字符，所以攻击者可能会在Wcb响应中提供soft或者seller，而实际上给恶意代码下达的是休眠的命令。流量分析可能确认攻击者在使用单词soft发送一个命令给恶意代码，而这可能会误导分析者在特征中使用完整单词。攻击者在不修改恶意代码的情况下，就可以无限制地使用seller或者任意以s开头的单词。

### 5.什么类型的编码用于命令参数？它与Base64编码有什么不同？它提供的优点和缺点各是什么？





sleep命令没有编码，而数字表示休眠的秒数。在其中的两条命令中，参数使用的是自定义编码，虽然简单但不是Base64编码。参数由偶数个数字来进行表示（一旦尾部的96被删除）每组两个数字代表的原始数字是数组 /abcdefghijklmnopgrstuvwxyz123456789:.的索引。这些参数仅用于URL间的通信，所以没有必要用到大写字符。这种编码方案的好处是：它不是标准算法，所以要理解它的内容，需要逆向工程分析它。缺点是很简单：在字符串输出中它可能被识别为可疑，因为URL总是以相同方式开头，这是一个一致性的模式。

### 6.这个恶意代码会接受哪些命令？

恶意代码命令包括quit、download、sleep 和redirectquit命令就是简单退出程序。download命令是下载并且运行可执行文件，不过与以前的实验不同，这里攻击者可以指定URL下载。redirect命令修改了恶意代码使用的配置文件，因此导致了 个新的信令URL。

### 7.这个恶意代码的目的是什么？

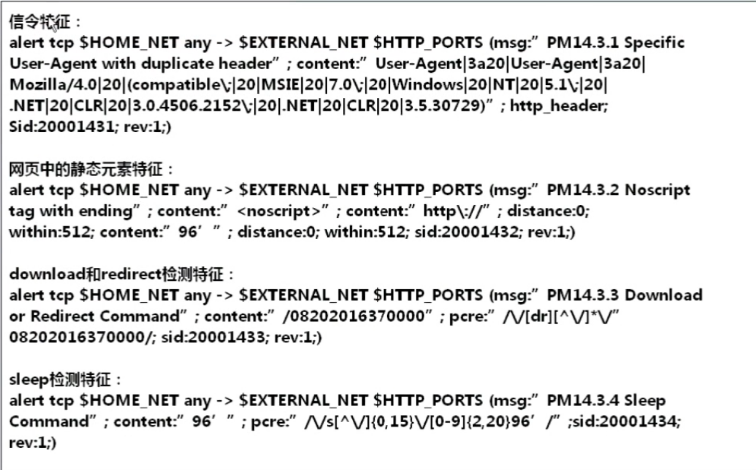
这个恶意代码本质上就是一个下载器。它有一些重要的优点，例如基于Web的控制，在确认为恶意域名并关闭后很容易做出调整。

### 8.本章介绍了用独立的特征，来针对不同位置代码的想法，以增加网络特征的鲁棒性。那么在这个恶意代码中，可以针对哪些区段的代码，或是配置文件，来提取网络特征？

恶意代码行为中某些特殊元素可能作为独立的检测目标，如下所示：

* 与静态定义的域名和路径，以及动态发现的URL中相似信息有关的特征。与信令中静态组件有关的特征。
* 能够识别命令初始请求的特征。
* 能够识别命令与参数对特定属性的特征。

### 9.什么样的网络特征集应该被用于检测恶意代码？



1. **实验结论及心得体会**

复习到课上的知识点，恶意代码的网络特征。

熟练使用恶意代码分析工具。