26 November 2022

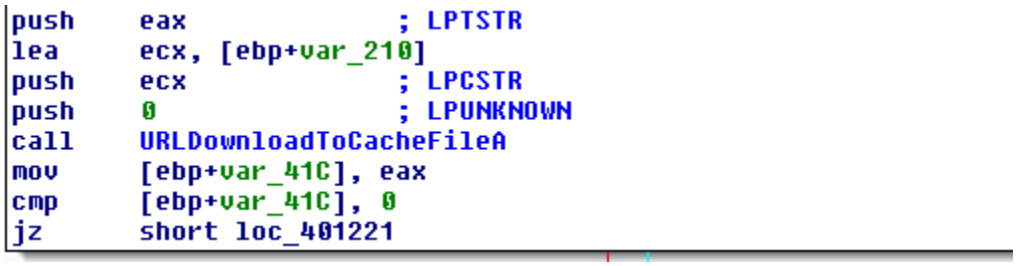
ASSIGNEMENT 11

LAB 14-1

Analyze the malware found in file Lab14-01.exe. This program is not harmful to your system.

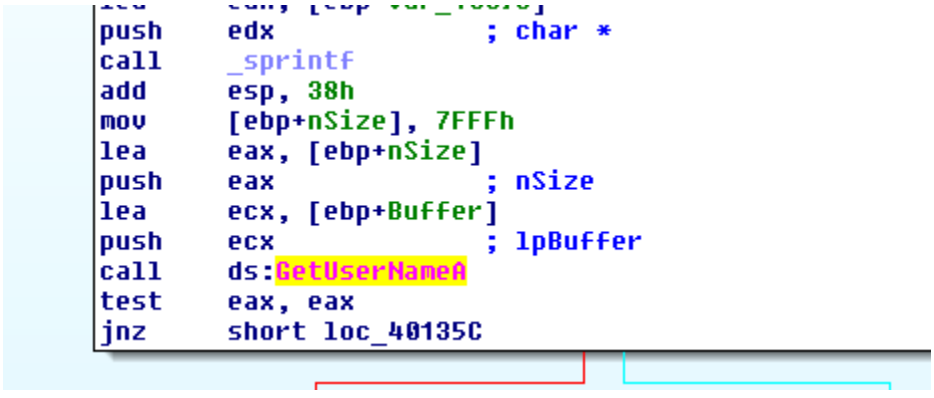
Questions

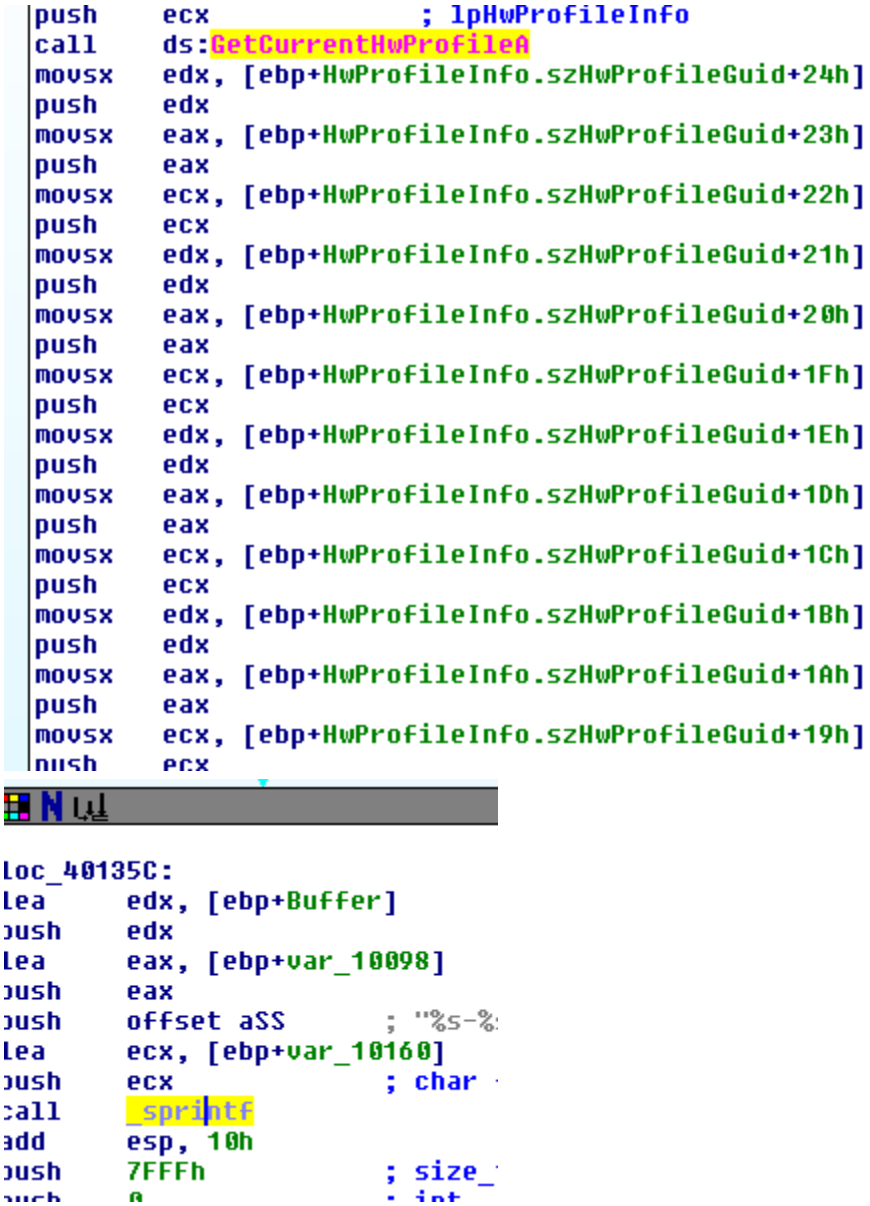
i. Which networking libraries does the malware use, and what are their advantages?



The application includes the COM interface based URLDownloadToCacheFile method. When malware exploits COM interfaces, the majority of the content of its HTTP requests originates from inside Windows, making network signatures ineffective. These requests will mix in with the normal ones that the infected host often makes.

ii. What source elements are used to construct the networking beacon, and what conditions would cause the beacon to change?





First, the virus calls GetCurrentHwProfile and extracts the last 12 bytes of the current user's hardware profile's 36-byte GUID. The function is then called GetUserName. Both are entered into a format string, with the username inserted first and the GUID inserted second. Several characters are placed onto the stack after GetCurrentHwProfileA; they will be utilized to construct a string with _sprintf.

iii. Why might the information embedded in the networking beacon be of interest to the attacker?

A decoding function exists on the Server that can decode the Path and so identify the infected host. The attacker is then able to transmit customized instructions to that host, which may vary from those provided to other hosts.

iv. Does the malware use standard Base64 encoding? If not, how is the encoding unusual?

The Base64 encoding is not standard because its padding utilizes an a instead of an equal sign. By replacing the last encoded letter with the standard padding character, the text may be completely decoded:

printf "ODA6NmQ6NjE6NzI6Njk6NmYtS2V2aW4" | base64 -d 80:6d:61:72:69:ff -Kevin" | base64 -d 80:6d:61:72:69:ff-Kevin

v. What is the overall purpose of this malware?

This is a downloading software. Using a GET request, the virus identifies itself to a host and downloads a PE executable file with a.png extension to the Internet cache. Then, the executable is executed. This virus downloads and runs additional programs.

vi. What elements of the malware's communication may be effectively detected using a network signature?

```
.data:00406030 ; char aHttpWww_practi[]
.data:00406030 aHttpWww_practi db 'http://www.practicalmalwareanalysis.com/%s/%c.png',0
.data:00406030                                         ; DATA XREF: sub_4011A3+39↑o
.data:00406062                 align 4
```

A basic network signature to verify the domain http://www.practicalmalwareanalysis.com/%s/%c.png might be good for detecting this malware, but is clearly unreliable in the long run since the domain can be simply altered by the malware author.
Targeted parts of the malware's communication include the domain name, the colons and dash found after Base64 decoding. The last character of the Base64 part of the URI is the single character used for the filename of the PNG file.

vii. What mistakes might analysts make in trying to develop a signature for this malware?

Defenders may attempt to target elements other than the URI if they are unaware that they are determined by the operating system. Typically, the Base64 string ends with an a, which results in the filename appearing as a.png. However, if the value of the username is an even multiple of three, the final

character and filename will rely on the final character of the encrypted username. In this instance, the filename is arbitrary.

viii. What set of signatures would detect this malware (and future variants)?

The first rule is a basic rule that detects requests to http://www.practicalmalwareanalysis.com, while the second rule is more precise and use a regular expression to identify beaconing requests. The domain name www.practicalmalwareanalysis.com was omitted since it may have been a hacked genuine site, but also because the command-and-control server can change.

LAB 14-2

Analyze the malware found in file Lab14-02.exe. This malware has been configured to beacon to a hard-coded loopback address in order to prevent it from harming your system, but imagine that it is a hard-coded external
address.

Questions

i. What are the advantages or disadvantages of coding malware to use direct IP addresses?

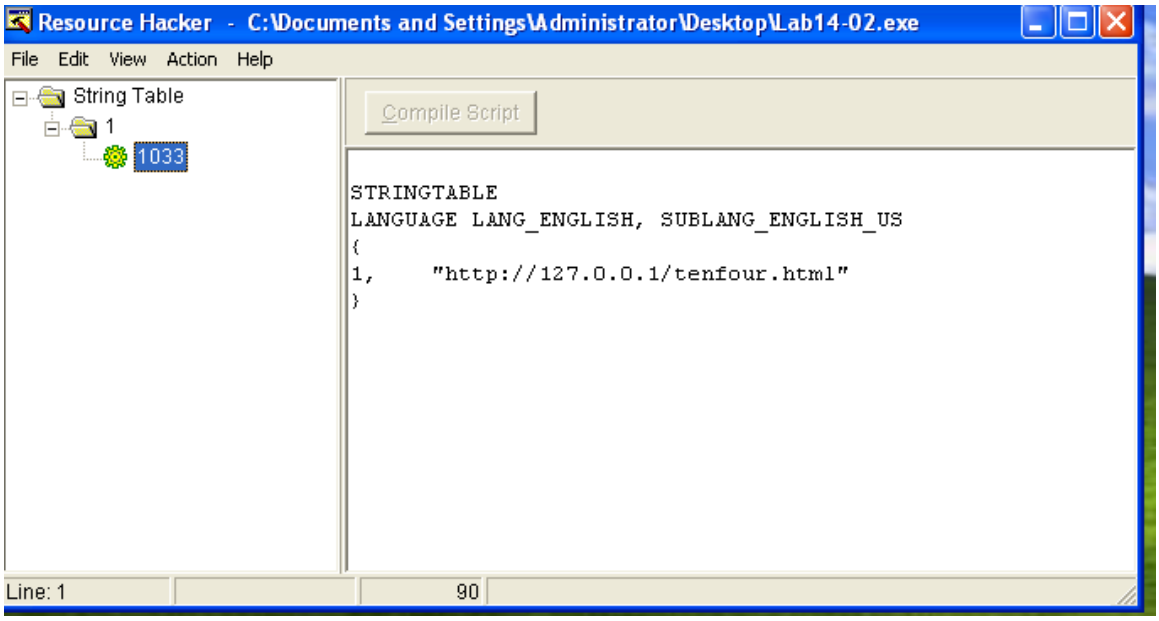
The attacker may find it more difficult to handle static IP addresses than domain names. Using DNS, an attacker is able to deploy his assets to any machine and dynamically reroute his bots by by changing a DNS address. The defender has several choices for establishing protections for both kinds of infrastructure, but for identical reasons, IP addresses are more difficult to manage than domain names. This alone might prompt an attacker to prefer static IP addresses over domains. If the IP of the attacker were banned, subsequent variants of the same virus that utilize a different IP would not be impacted. The attacker would have lost access to the virus if the IP is blacklisted as malicious and prohibited by the government. If the attacker uses a domain name, he may simply redirect to a different IP address.

ii. Which networking libraries does this malware use? What are the advantages or disadvantages of using these libraries?

.rdata:0...	00000010	C	ShellExecuteExA
.rdata:0...	0000000C	C	SHELL32.dll
.rdata:0...	00000014	C	InternetCloseHandle
.rdata:0...	00000011	C	InternetOpenUrlA
.rdata:0...	0000000E	C	InternetOpenA
.rdata:0...	00000011	C	InternetReadFile
.rdata:0...	0000000C	C	WININET.dll

Malware employs WinINet libraries. These libraries have the drawback of requiring a hard-coded User-Agent and, if required, additional headers. The WinINet packages have a benefit over the Winsock API. The inflexibility of higher-level libraries is a drawback.

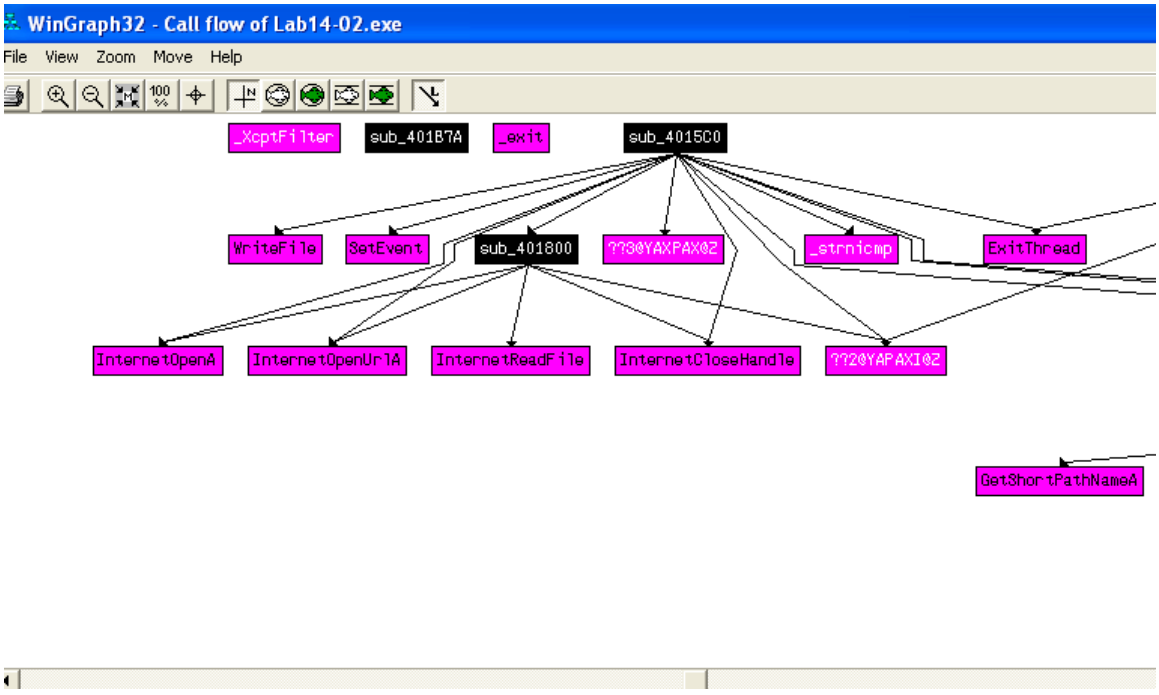
iii. What is the source of the URL that the malware uses for beaconing? What advantages does this source offer?



VA	Raw Data																Value
00401800	00	00	00	00	00	00	00	00	04	00	00	00	00	00	01	00	. . . . .
00401810	06	00	00	00	18	00	00	80	00	00	00	00	00	00	00	00	. . . . .
00401820	04	00	00	00	00	00	01	00	01	00	00	00	30	00	00	80	. . . . . 0 . . .
00401830	00	00	00	00	00	00	00	00	04	00	00	00	00	00	01	00	. . . . .
00401840	09	04	00	00	48	00	00	00	58	40	00	00	5A	00	00	00	. . . . H . . . X @ . . Z . . .
00401850	E4	04	00	00	00	00	00	00	00	00	1D	00	68	00	74	00	. . . . . h . t .
00401860	74	00	70	00	3A	00	2F	00	2F	00	31	00	32	00	37	00	t . p . : . / . / . 1 . 2 . 7 .
00401870	2E	00	30	00	2E	00	30	00	2E	00	31	00	2F	00	74	00	. . 0 . . . 0 . . . 1 . / . t .
00401880	65	00	6E	00	66	00	6F	00	75	00	72	00	2E	00	68	00	e . n . f . o . u . r . . . h .
00401890	74	00	6D	00	6C	00	00	00	00	00	00	00	00	00	00	00	t . m . l . . . . .
004018A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
004018B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
004018C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
004018D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
004018E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
004018F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
00401900	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .
00401910	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	. . . . .

The resource area allows the attacker to deliver several backdoors to various command-and-control servers without recompiling the malware. The source of the URL beaconed by the virus is http://127.0.0.1/tenfour.html. There is a 16-bit unicode string due to the null byte after each character.

iv. Which aspect of the HTTP protocol does the malware leverage to achieve its objectives?



```
.data:0040306C ; char szAgent[]
.data:0040306C szAgent          db 'Internet Surf',0    ; DATA XREF: sub_401800+B↑o
.data:0040307A                  align 4
.data:0040307C aOpen            db 'Open',0             ; DATA XREF: sub_401880+C6↑o
.data:00403081                  align 4
.data:00403084 ; char aNul[]
.data:00403084 aNul             db ' > nul',0           ; DATA XREF: sub_401880+91↑o
.data:0040308B                  align 4
.data:0040308C ; char String2[]
.data:0040308C String2          db '/c del ',0          ; DATA XREF: sub_401880+69↑o
.data:00403094 ; char Name[]
.data:00403094 Name             db 'COMSPEC',0          ; DATA XREF: sub_401880+4F↑o
```

The attacker exploits the HTTP User-Agent field, which should provide information about the application. The virus generates one thread that encodes outgoing data in this field and another that utilizes a static field to signify that it is the "receiving" side of the channel. Malware generates two separate threads. One can deliver data encoded with a custom base64 string in the user agent field. The second thread that receives and reads data employs a static user agent called Internet Surf.

v. What kind of information is communicated in the malware's initial beacon?

After determining that InternetOpen was being utilized, the user agent is populated. This function appears to be encoded and accepts two buffers as arguments. Which was called just before the InternetOpen function, so it looked like a decent choice for encoding.

vi. What are some disadvantages in the design of this malware's communication channels?

Although the attacker encrypts departing data, he does not encode incoming orders. In addition, since the server must differentiate between the two communication channels based on the static contents of the User-Agent fields, this server reliance is obvious and may be targeted using signatures.

vii. Is the malware's encoding scheme standard?

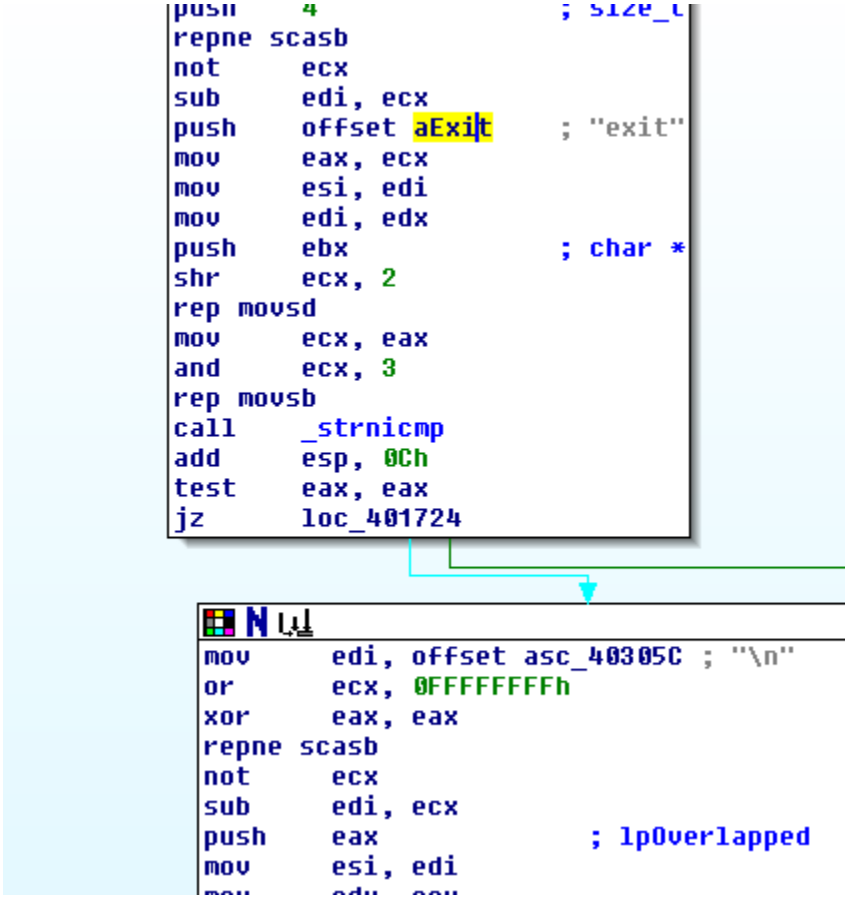
```
.data:00403010 byte_403010      db 57h                  ; DATA XREF: sub_401000+80↑r
.data:00403010                                          ; sub_401000+88↑r ...
.data:00403011 aXyzlabcd3fghij db 'XYZlabcd3fghijko12e456789ABCDEFGHIJKL+/MNOPQRSTUVmn0pqrstuvwxyz',0
.data:00403051                  align 4
.data:00403054 aCmd_exe         db 'cmd.exe',0          ; DATA XREF: WinMain(x,x,x,x)+13D↑o
.data:0040305C asc_40305C:                              ; DATA XREF: sub_4015C0+E2↑o
.data:0040305C                  dw 0Ah
.data:0040305C                  unicode 0, <>,0
.data:00403060 ; char aExit[]
.data:00403060 aExit            db 'exit',0             ; DATA XREF: sub_4015C0+BA↑o
```

No, the malware's encoding scheme is not standard due to creating a custom string. The custom is the following:
XYZlabcd3fghijko12e456789ABCDEFGHIJKL+/MNOPQRSTUVmn0pqrstuvwxyz

Crating a python script one can decode the string, after decoding one can find discover the following:

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

viii. How is communication terminated?



Using the keyword exit, communication is ended. The virus attempts to remove itself upon termination.

ix. What is the purpose of this malware, and what role might it play in the attacker's arsenal?

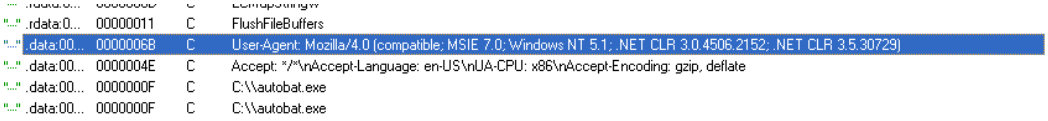
This malicious software is a reverse command shell that an attacker may use to execute arbitrary commands on affected devices. The software attempts to erase itself upon connection failure or termination, hence it is likely intended for a single usage.

LAB 14-3

This lab builds on Lab 14-1. Imagine that this malware is an attempt by the attacker to improve his techniques. Analyze the malware found in file Lab14-03.exe

Questions

i. What hard-coded elements are used in the initial beacon? What elements, if any, would make a good signature?



Accept, Accept-Language, UA-CPU, Accept-Encoding, and User-Agent are hard-coded headers. User-Agent: User-Agent: Mozilla is duplicated due to the erroneous inclusion of an extra User-Agent: by the malware author. The whole User-Agent header constitutes an efficient signature.

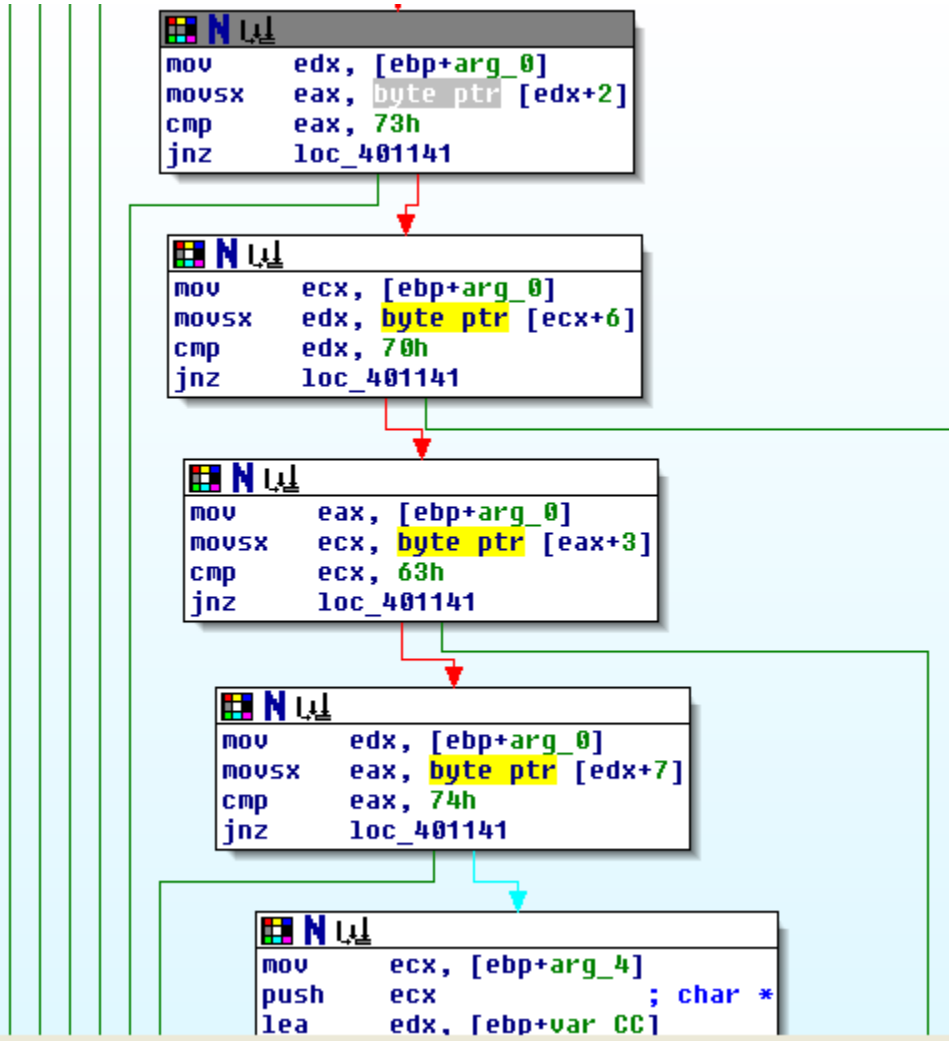
ii. What elements of the initial beacon may not be conducive to a longlasting signature?

Only when the configuration file is missing are the domain name and path of the URL hard-coded. Signatures must be generated for this URL as well as any configuration files seen. Using http://www.practicalmalwareanalsysis.com/start.htm as a signature may not be a good option since an attacker might alter the beacon destination.

iii. How does the malware obtain commands? What example from the chapter used a similar methodology? What are the advantages of this technique?

The virus retrieves a URL from the file C:autobat.exe and parses the HTML content for instructions after a <noscript> tag .
This approach has the benefit of being stealthy. It is difficult to identify encoded instructions concealed inside application data, particularly if the application data is valid. Additionally, an attacker may simply modify the HTML content on the server in order to execute new programs and update the infection. The client may simply modify the URL, allowing the attacker to switch servers and command pages.

iv. When the malware receives input, what checks are performed on the input to determine whether it is a valid command? How does the attacker hide the list of commands the malware is searching for?



The directives are concealed during transmission by being a single byte, which is the initial character of the URL's path. The fact that anything may follow the

initial byte of a command makes it seem harmless and diverse. As the second directory in the URL route, an argument is sent. The check for the <noscript> tag output sequence is out of order inside the executable, impeding string analysis and requiring the analyzer to manually reassemble the string.

v. What type of encoding is used for command arguments? How is it different from Base64, and what advantages or disadvantages does it offer?

```
.rdata:004070D4
.rdata:004070D4 ; Segment type: Pure data
.rdata:004070D4 ; Segment permissions: Read
.rdata:004070D4 _rdata          segment para public 'DATA' use32
.rdata:004070D4                 assume cs:_rdata
.rdata:004070D4                 ;org 4070D4h
.rdata:004070D4                 align 8
.rdata:004070D8 byte_4070D8     db 2Fh                  ; DATA XREF: sub_401147+7E↑r
.rdata:004070D9                 db  61h ; a
.rdata:004070DA                 db  62h ; b
.rdata:004070DB                 db  63h ; c
.rdata:004070DC                 db  64h ; d
.rdata:004070DD                 db  65h ; e
.rdata:004070DE                 db  66h ; f
.rdata:004070DF                 db  67h ; g
.rdata:004070E0                 db  68h ; h
.rdata:004070E1                 db  69h ; i
.rdata:004070E2                 db  6Ah ; j
.rdata:004070E3                 db  6Bh ; k
.rdata:004070E4                 db  6Ch ; l
.rdata:004070E5                 db  6Dh ; m
.rdata:004070E6                 db  6Eh ; n
.rdata:004070E7                 db  6Fh ; o
.rdata:004070E8                 db  70h ; p
.rdata:004070E9                 db  71h ; q
.rdata:004070EA                 db  72h ; r
.rdata:004070EB                 db  73h ; s
.rdata:004070EC byte_4070EC     db 74h                  ; DATA XREF: __output+4A↑r
.rdata:004070ED aUvwxyz01234567 db 'uvwxyz0123456789:.',0
.rdata:00407100 unk_407100      db 0FFh                 ; DATA XREF: start+5↑o
.rdata:00407101                 db 0FFh
.rdata:00407102                 db 0FFh
.rdata:00407103                 db 0FFh
```

There is no encoding for the sleep command; the number indicates the duration of sleep in seconds. For two of the instructions, the argument is encoded using a non-Base64 encoding, albeit a basic one. Argument given with an even number of numbers (once the trailing 96 is removed). Each pair of two digits represents the raw integer that is an index into the array /abcdefghijklmnopqrstuvwxyz0123456789:.

A benefit of the encoding is that because it is bespoke, it lacks standard libraries and needs more effort on the side of the analyst. A downside is that the encoding is quite straightforward.

vi. What commands are available to this malware?

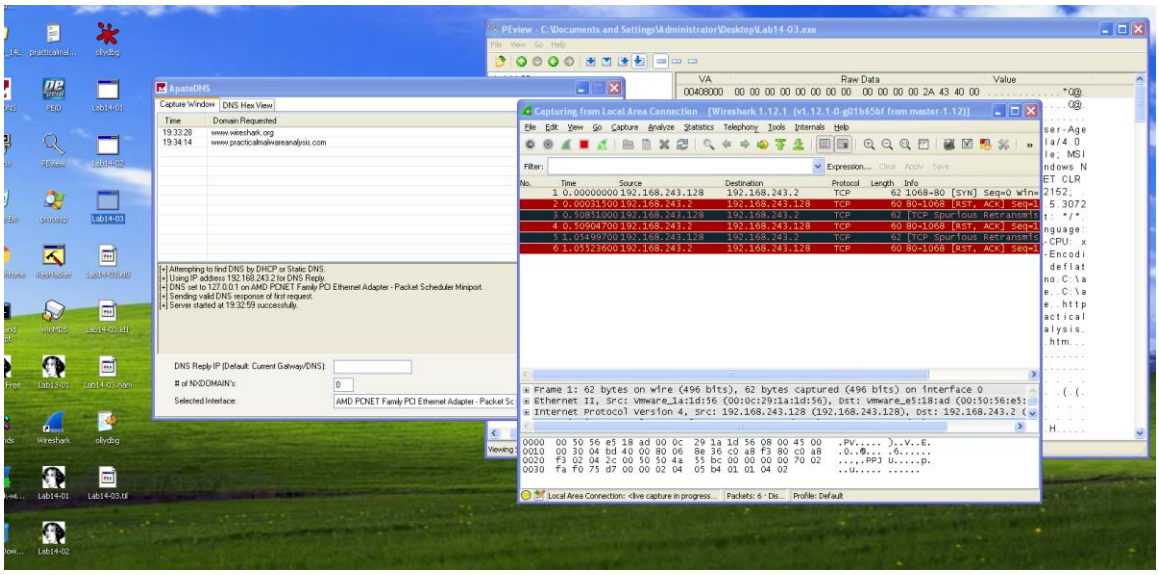
Included among the malware commands are quit, download, sleep, and redirect. The quit command simply terminates the application. The download command

downloads and executes an executable, however unlike the previous lab, the attacker may provide the URL from where to get the executable, etc.

vii. What is the purpose of this malware?

Included among the malware commands are quit, download, sleep, and redirect. The quit command simply terminates the application. The download command downloads and executes an executable, however unlike the previous lab, the attacker may provide the URL from where to get the executable, etc.

viii. This chapter introduced the idea of targeting different areas of code with independent signatures (where possible) in order to add resiliency to network indicators. What are some distinct areas of code or configuration data that can be targeted by network signatures?



Network signatures may be used to target the initial beacon, hardcoded HTTP headers, duplicate User-Agent headers, and the layout of the HTML response body command. There seemed to be nothing wrong with the HTTP headers in the strings output, and the HTTP headers in the packet capture were okay at first sight. Some independently targets may be signatures associated with the statically specified domain and path, as well as comparable information from any dynamically found URLs. Signatures associated with the beacon's static components, signatures that provide the first command needs, and signatures that identify certain command and argument pair properties.

ix. What set of signatures should be used for this malware?

Some signatures that should be used, http://www.practicalmalwareanalsysis.com/start.htm, any url found in c:\\autobat.exe, headers(ex: user agent), and http response that contain <noscript>