

Proj 16: Data Encoding (Lab 13-1) (25 pts.)

What you need:

- A Windows machine with the tools we have been using installed. I did this project easily on Win 10 TP. It also works on Windows Server 2008.

Purpose

You will practice the techniques in chapter 13.

Beacons

The book recommends running the malware with another VM simulating the Internet with inetsim, but I don't see any good reason to bother with that. I just connected a VM to the real Internet and ran the malware.

Launch the **Lab13-01.exe** file.

Use either method, and capture a beacon with Wireshark.

Adjust the wireshark window to show these two features, highlighted below:

- GET /randomletters/ HTTP/1.1**
- Host: www.practicalmalwareanalysis.com**

Save this image with the filename "**Proj 16a from YOUR NAME**".

YOU MUST SUBMIT WHOLE-DESKTOP IMAGES TO GET FULL CREDIT!



Wireshark 1.10.1 (SVN Rev 50926 from /trunk-1.10))

Filter: http

No.	Time	Source	Destination	Protocol	Length	Info
22	0.30958100	192.168.119.167	192.0.81.250	HTTP	150	GET /v0l0wFBTUDM=/ HTTP/1.1
24	0.36971600	192.0.81.250	192.168.119.167	HTTP	449	HTTP/1.1 301 Moved Permanently
28	0.46980100	192.0.81.250	192.168.119.167	HTTP	449	[TCP Retransmission] HTTP/1.1
36	0.62121300	192.168.119.167	192.0.81.250	HTTP	170	GET /v0l0wFBTUDM=/ HTTP/1.1
60	1.12181200	192.0.81.250	192.168.119.167	HTTP	122	HTTP/1.1 404 Not Found (text/

Frame 22: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits) on interface 0
 Ethernet II, Src: Vmware_60:b4:15 (00:0c:29:60:b4:15), Dst: Vmware_e3:22:f1 (00:50:56:e3:22:f1)
 Internet Protocol Version 4, Src: 192.168.119.167 (192.168.119.167), Dst: 192.0.81.250 (192.0.81.250)
 Transmission Control Protocol, Src Port: sgi-storman (1178), Dst Port: http (80), Seq: 1, Ack: 1,
 Hypertext Transfer Protocol
 GET /v0l0wFBTUDM=/ HTTP/1.1\r\n
 User-Agent: Mozilla/4.0\r\n
 Host: www.practicalmalwareanalysis.com\r\n
 \r\n
 [Full request URI: http://www.practicalmalwareanalysis.com/v0l0wFBTUDM=/]
 [HTTP request 1/2]
 [Response in frame: 24]

0000 00 50 56 e3 22 f1 00 0c 29 60 b4 15 08 00 45 00 .PV."...)....E.
 0010 00 88 09 55 40 00 80 06 a6 d0 c0 a8 77 a7 c0 00 ...U@...w...
 0020 51 fa 04 9a 00 50 91 9f 30 96 58 2a ed 75 50 18 Q....P.. 0.X".uP.
 0030 fa f0 51 49 00 00 47 45 54 20 2f 56 30 6c 4f 57 ..QI..GE T /v0l0w
 0040 46 42 54 55 44 4d 3d 2f 20 48 54 54 50 2f 31 2e FBTUDM=/ HTTP/1.
 0050 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d l..User- Agent: M
 0060 6f 7a 69 6c 6c 61 2f 34 2e 30 0d 0a 48 6f 73 74 ozilla/4.0 Host

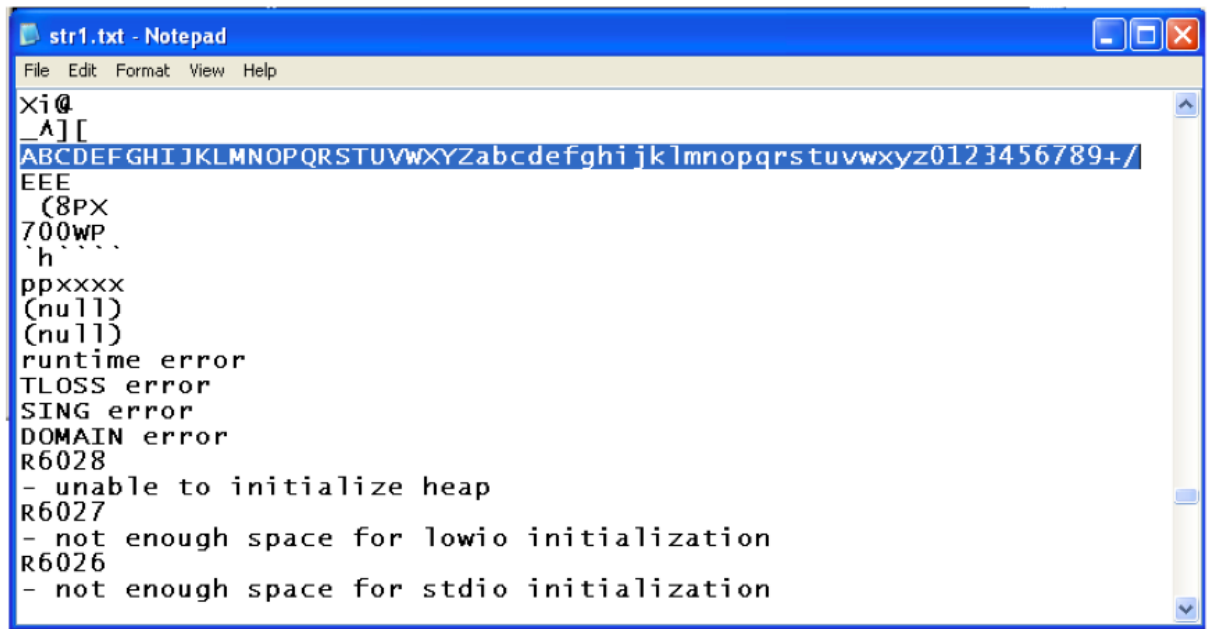
Frame (frame), 150 bytes Packets: 124 · Displayed: 5 (4.0%) Profile: Default

Strings

The book uses strings only to point out two strings that are not visible. But there is an interesting string present, showing the use of Base64 encoding.

Examine the strings in the **Lab13-01.exe** file.

Save an image showing the string highlighted below, with the filename "**Proj 16b from YOUR NAME**".



IDA Pro

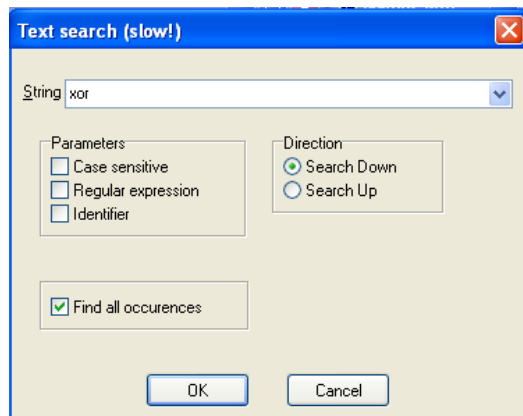
Open **Lab13-01.exe** file in IDA Pro.

Click **Options, General**. Check "**Line Prefixes**" and click **OK**.

Click in the "IDA View-A" window to make it active.

From the menu bar, click **Search, text....**

In the Text Search dialog, enter **xor** and check "**Find all occurrences**", as shown below:



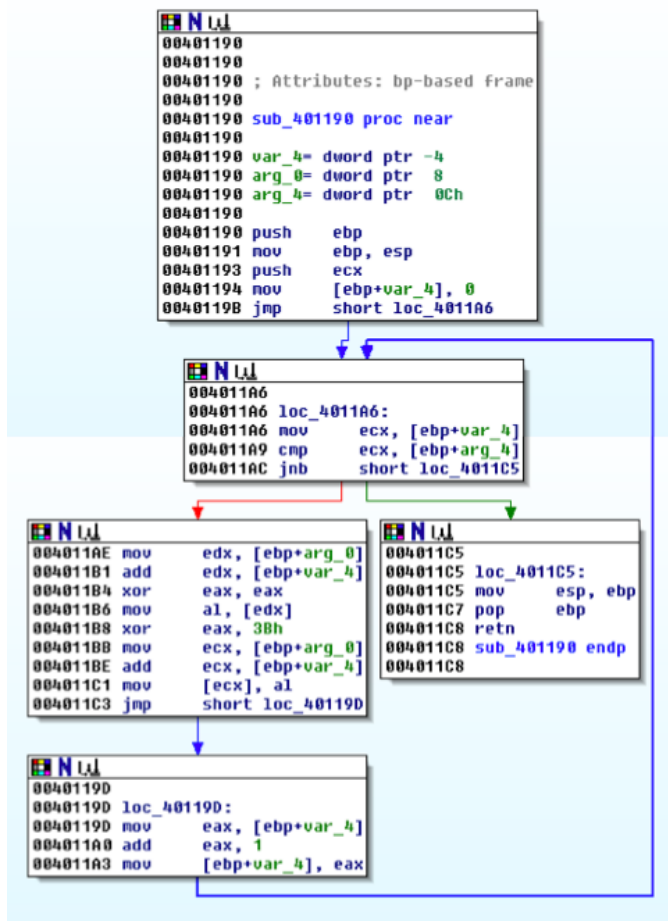
A list of locations using the XOR command appears, as shown below.

Occurrences of: xor	
Address	Instruction
.text:00401007	xor ecx, ecx
.text:0040101C	xor edx, edx
.text:00401029	xor ecx, ecx
.text:0040104E	xor eax, eax
.text:0040105C	xor edx, edx
.text:0040108D	xor ecx, ecx
.text:004011B4	xor eax, eax
.text:004011B8	xor eax, 3Bh
.text:004011D6	xor eax, eax
.text:004012A2	xor al, al
.text:004012E6	xor al, al
.text:004012FA	xor al, al
.text:00401332	xor eax, eax
.text:00401350	xor eax, eax
.text:0040138E	xor eax, eax
.text:00401463	xor eax, eax
.text:004021E5	xor ecx, ecx
.text:00402202	xor edx, edx
.text:00402BE2	xor dh, [eax]
.text:00402BE6	xor [eax], dh

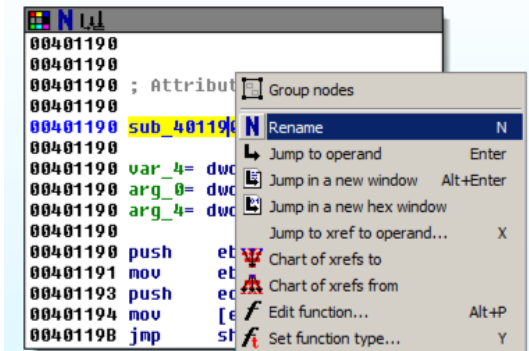
Double-click the **xor eax, 3Bh** instruction.

You should see the function shown below.

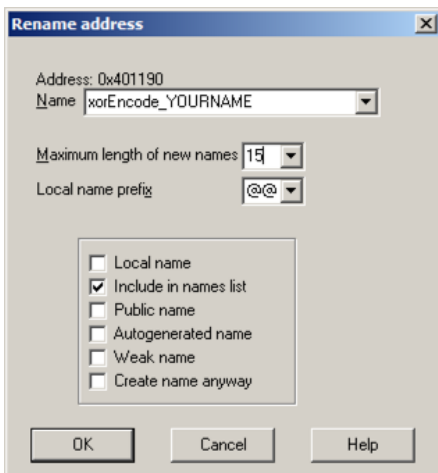
As explained in the book, this function performs xor encoding.



In the top box of the function, right-click **sub_401190** and click **Rename**, as shown below.



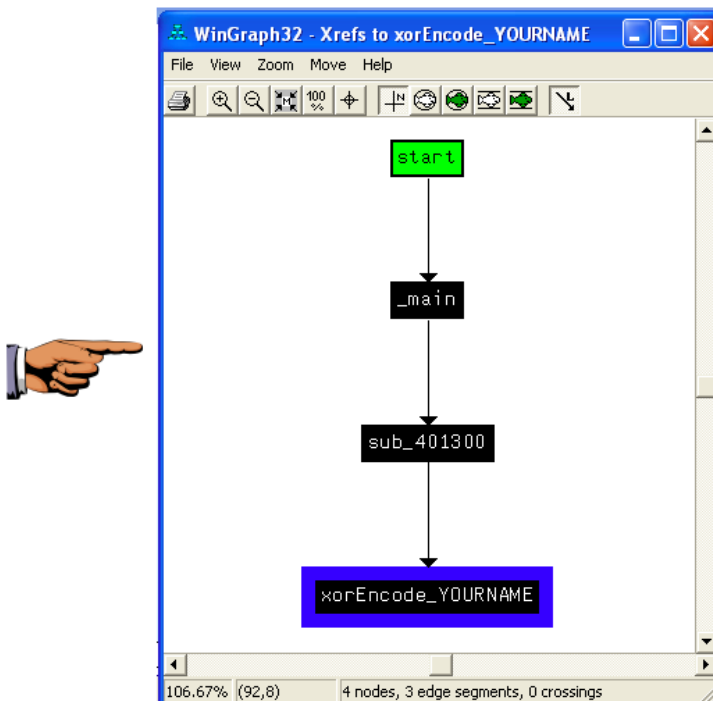
Enter a new name of `xorEncode_YOURNAME`, as shown below, replacing "YOURNAME" with your own name.



Click **OK**. If you are prompted to, increase the name length limit.

Right-click `xorEncode_YOURNAME` and click "Chart of xrefs to".

A chart showing four boxes appears, ending with one containing your name, as shown below.



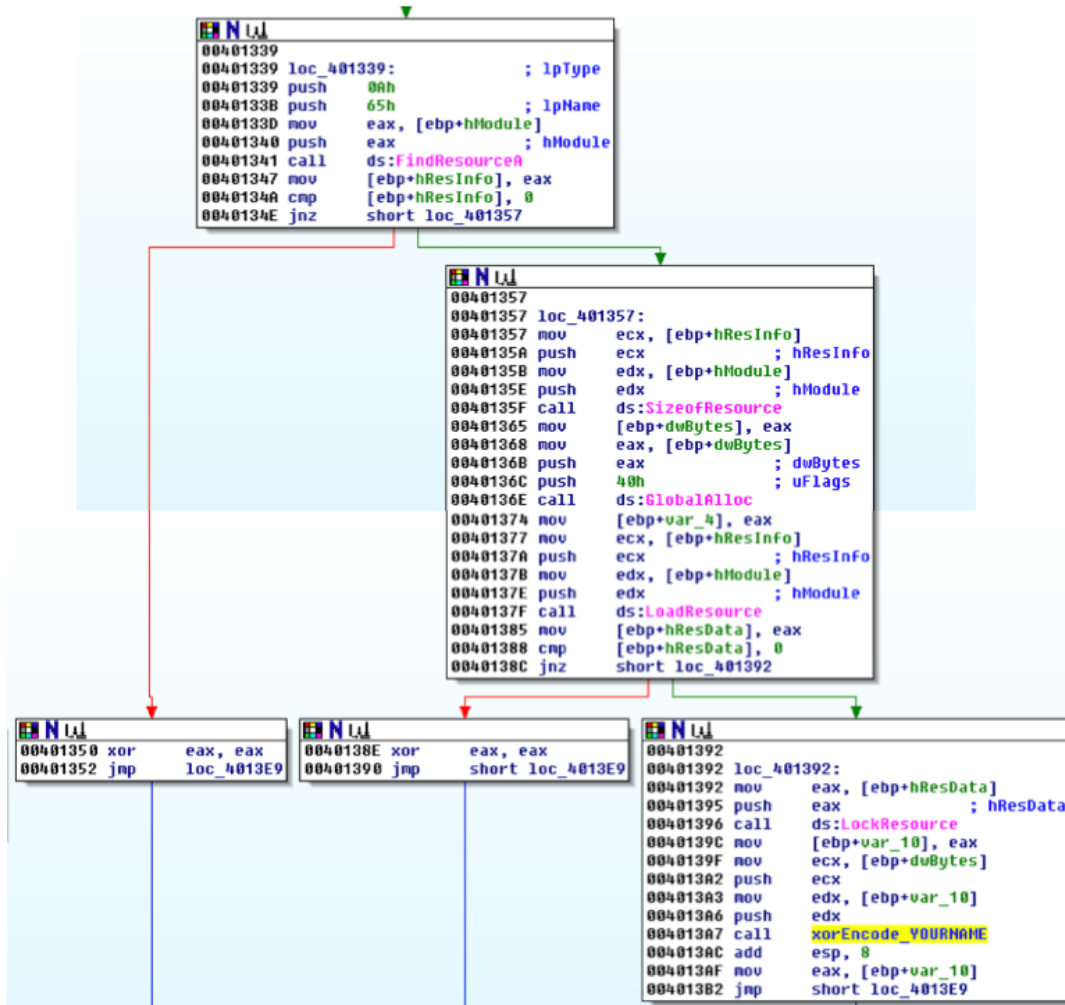
Save an image showing the **four boxes** with **Your name** in the bottom one, "**Proj 16c from YOUR NAME**". Close the "WinGraph32 - Xrefs to xorEncode..." box.

Right-click `xorEncode_YOURNAME` and click "Jump to xref to operand...".

A box pops up showing the address of the xref. Click **OK**.

This function, as shown below, calls these functions (shown in pink letters):

- FindResourceA
- SozeofResource
- GlobalAlloc
- LoadResource
- LockResource



As explained in the book, this code loads a resource and then encodes it.

The resource is identified by its index of 65h, specified in the code at location 401338.

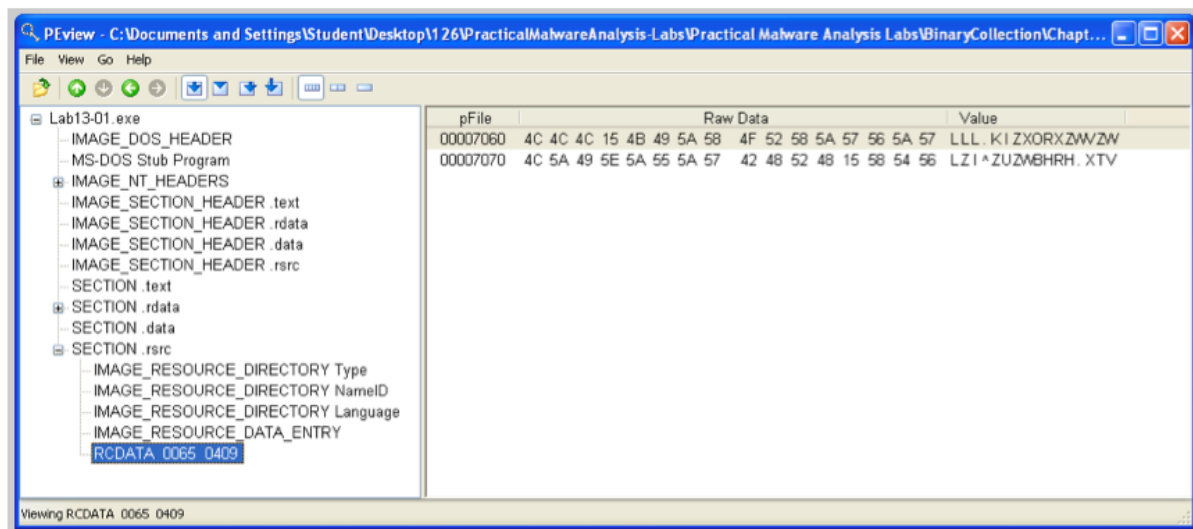
PEview

Open the **Lab13-01.exe** file in PEview.

In the left pane, click the **RCDATA 0065 0409** resource.

In the right pane, find the starting address **00007060**, as shown below.

Save an image showing **RCDATA 0065 0409** and **00007060** with the filename "**Proj 16d from YOUR NAME**".



WinHex

In a Web browser, go to:

<http://winhex.com/winhex/>

On the left side, click the **Download** button, as shown below.



Right-click the **winhex.zip** file, click "Extract All", and click **Extract**.

A window appears showing the files contained in the winhex archive. Double-click **setup.exe**. Accept the default options to install WinHex. When the installation is complete, WinHex runs.

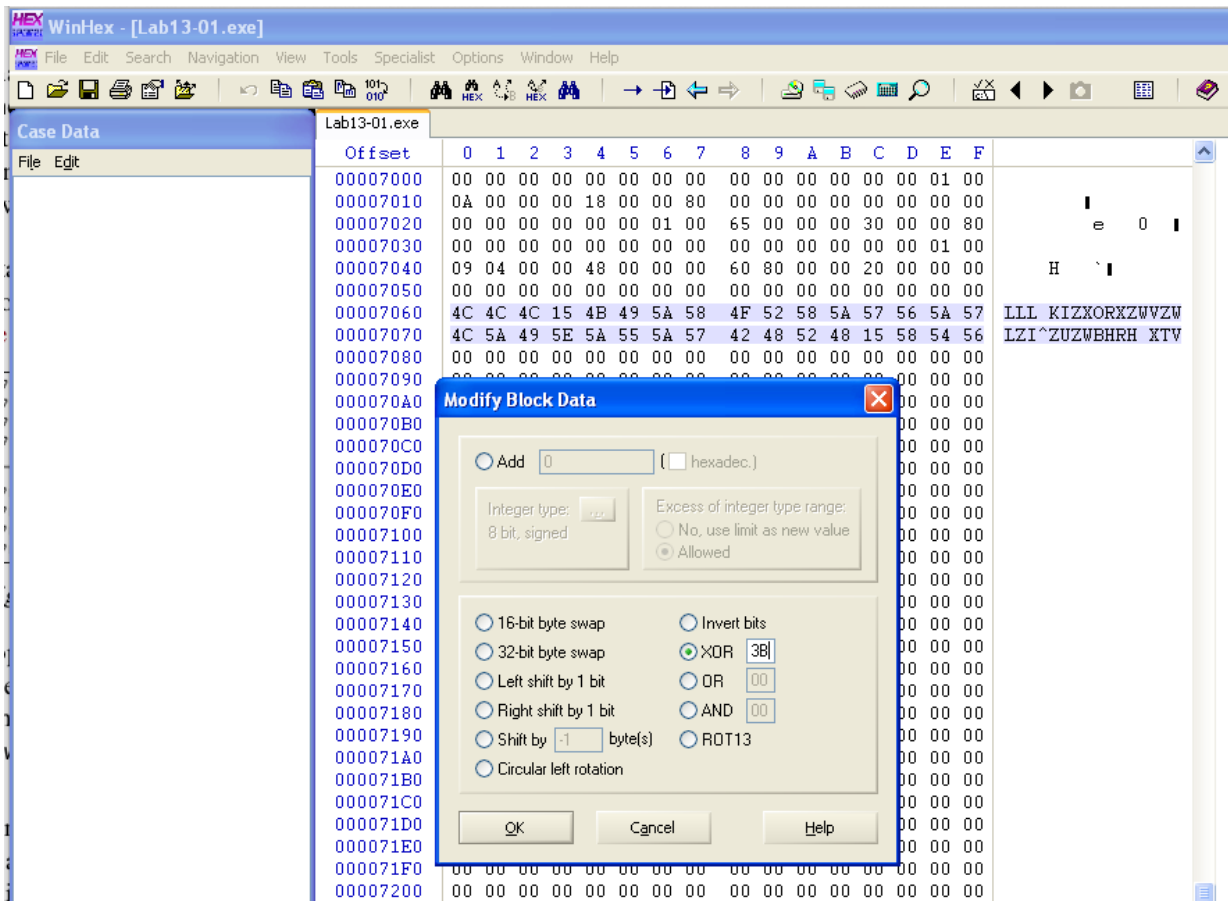
Troubleshooting

If double-clicking setup.exe does nothing, open a Command Prompt and launch it from there.

In WinHex, click **File, Open**. Open the **Lab13-01.exe** file in WinHex. Highlight bytes 7060 through 707F, as shown below.

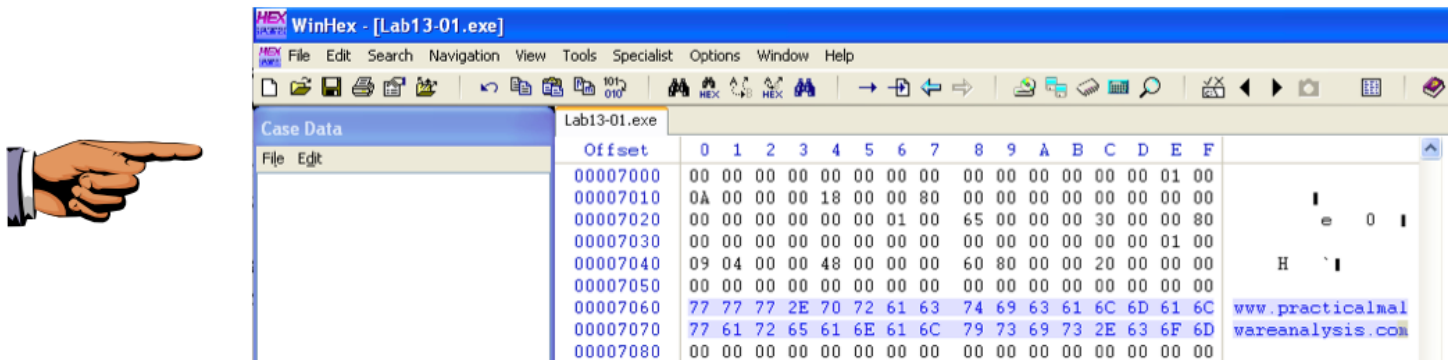
Click **Edit, "Modify Data"**.

In the "Modify Block Data" box, check the **XOR** radio button and enter a key of **3B**, as shown below:



Click **OK**.

The decoded string appears on the right side: "www.practicalmalwareanalysis.com", as shown below:



Save an image showing www.practicalmalwareanalysis.com with the filename "Proj 16e from YOUR NAME".

Turning in your Project

Email the images to cnit.126sam@gmail.com with the subject line: **Proj 16 from YOUR NAME**

Last modified 4-24-17