Exploit Compiling, *Writing*, and Deployment

Instructor Note: The intent of this class is for MIDN to find, download, compile, modify/write, and deploy and exploit. The exploit will initially be in the form of a C program. The MIDN will use the instructions below to complete the assignment. Completing the step-by-step instructions below should earn the MIDN a maximum of 85% on the lab. That is, they should get 85% for copying what we have provided for them. The other 15% is awarded at the discretion of the instructor by considering the attempt, and/or success of the MIDN successfull deploying the exploit. Details are included below. This is not a trivial task and is more complex than what is expected of MIDN at this point in time. Assistance from the Professor should be minimal if at all. MIDN may use any source to determine the solution as long as they cite it.

Overview

Cross-compilation allows you to develop for one platform (like Kali Linux) and compile to run on a different platform (such as Windows). For developers, it means that they can work on their platform of choice and compile their code for their target platform. For offensive cyber operators, it means we can compile exploit code for Windows from Kali.

The majority of available offensive cyber tools are targeted towards Linux, assuming most cyber operators will be operating on some form of Linux environment. Most Linux distros include a compilation toolchain, which will happily build local exploit code for Linux targets all day. If the machine we're targeting is Windows, Metasploit has plenty of Windows exploits, but this leaves us unable to compile any Windows exploits ourselves. This is where cross-compilation comes into play.

In this lab, we will go through the installation and basic usage of MinGW-w64 on Kali Linux to compile local exploit code. The MinGW-w64 project is a complete runtime environment for GCC (GNU Compiler Collection) to support binaries native to Windows 64-bit and 32-bit operating systems.

For this lab, MIDN will receive 85% for successfully **following** the step-by-step instructions below, taking the appropriate screenshots when necessary, and submitting the PDF. MIDN will receive an additional 5% for attempting, and detailing the steps they take, to modify the exploit code (i.e., C code). To be clear, an attempt is defined by modifying the exploit code provided in a manner that you believe will successfully execute on target, and provide you a shell to execute commands (i.e., ipconfig). MIDN should explain in detail the steps they take to accomplish this task. Finally, MIDN will receive the additional 10% for successfully modifying the exploit code, and obtaining a shell on the target. MIDN should include the aforementioned details and provide step-by-step screenshots of the successful exploit/shell/ipconfig.

MIDN may use <u>any</u> sources as long as they cite them.

MIDN will complete the lab by following these high-level steps:

- Review the step-by-step instructions below
- Launch Kali Linux Cyber Operator VM (SY401_Kali_alpha) and Windows XP VM (SY401_GroupX_WinXP)
- Execute the step-by-step instructions below
- <u>Capture all screenshots</u> of **your** Kali and Windows VM hosts as is provided below, and deliver the step-by-step process via a PDF **(+85%)**
- Research the exploit provided in the example, or another of your choice to modify and successfully execute (sources provided below)

- Modify the exploit source code
- Compile the exploit source code
- In addition to the above deliverable, provide a **screen capture** of your source code
- If your exploit is unsuccessful in obtaining a shell on the host, describe what you think may be the issue **(+5%)**
- Take a screenshot of the Meterpreter shell once you have successfully exploited the Target VM with your modified code **(+10%)**

MIDN should read the tool documentation, installation guides, and perform Internet searches to find solutions to challenges they encounter. The Professor will provide minimal assistance. MIDN can use the following documentation to start:

- How to Write Remote Exploits
- Buffer Overflow Exploit Example
- Writing Exploits for Win32 Systems from Scratch
- Lessons Learned Writing Exploits
- MinGW-w64 - for 32 and 64 bit Windows
- Nano Basics
- SearchSploit
- Exploit-DB
- Security Focus

Lab Deliverables

MIDN will submit a single PDF document to your Professor that contains the screenshots described above as your deliverable. The screenshots should be properly labeled. It is suggested that MIDN insert each of the required screenshots into a Microsoft Word document and export to a .PDF file.

For Sections 1131 and 5531 - MIDN should submit their file into their own folder in their assigned section folder in the SY401 Shared Folder found here If you do not have a folder, please create a folder with your last name under your assigned section and place your lab deliverable in that folder. If your professor prefers email submissions - the subject line of the email should be in the following format:

The subject line of the email should be in the following format:

```
SY401 [Section Number]: [NAME OF LAB] (alpha)
```

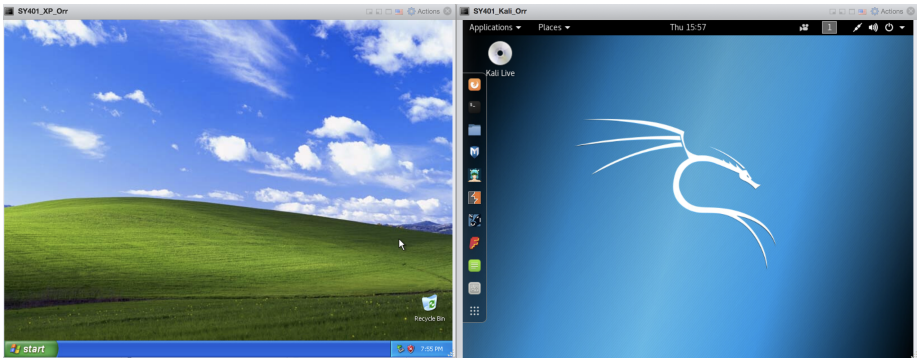
For example:

```
SY401 1111: Lowering the Barrier to Entry - Open Source Tools (m123456)
```

MIDN should gracefully shutdown their Virtual Machines (VMs) at the end of class, or whenever they are not using them. Failing to do so will result in a non-graceful shutdown from SY401 Faculty each day. Students risk losing work if this simple process is not followed.

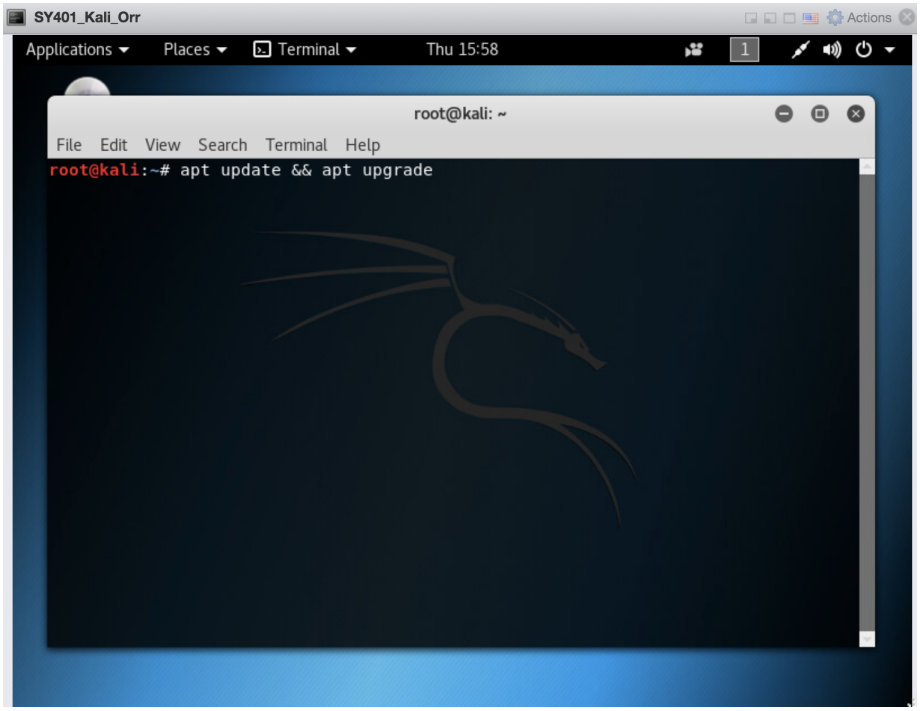
Cross Compiling Exploits

MIDN should launch the Kali Linux Cyber Operator VM (SY401_Kali_alpha) and the Windows XP VM (SY401_XP_alpha). Obtain the IP address of each system and remember for use in later steps.



In the Kali Linux VM, update available packages and upgrade out-of-date packages on our system. This can be done by typing

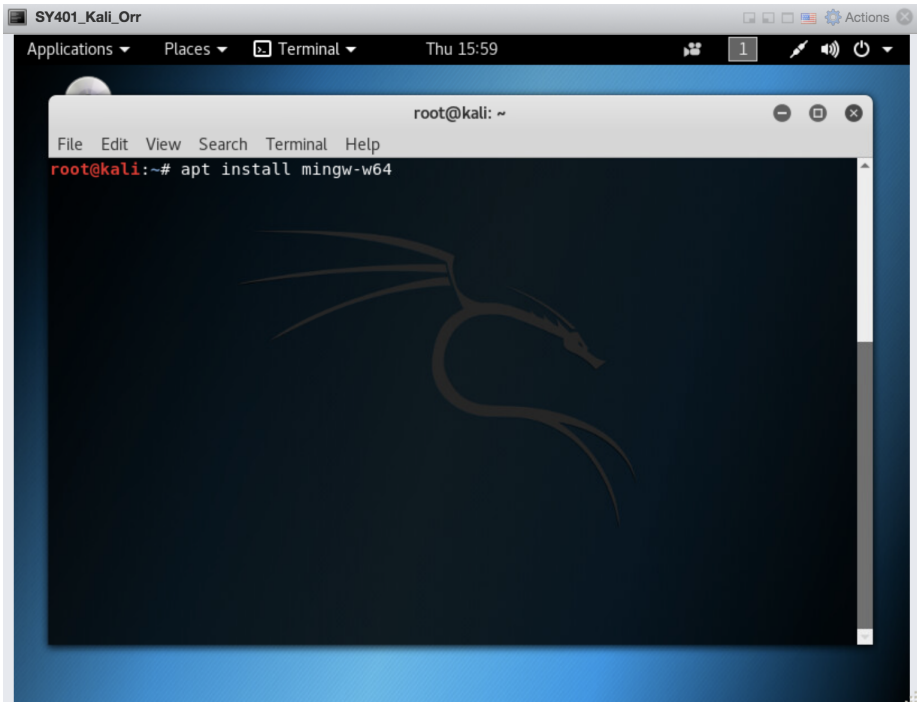
```
apt update && apt upgrade
```



Once your package lists are updated and the system is upgraded, it's time to install MinGW-w64. This can be done by typing:

```
apt install mingw-w64
```

Kali will prompt you to confirm the installation. Press enter to proceed.



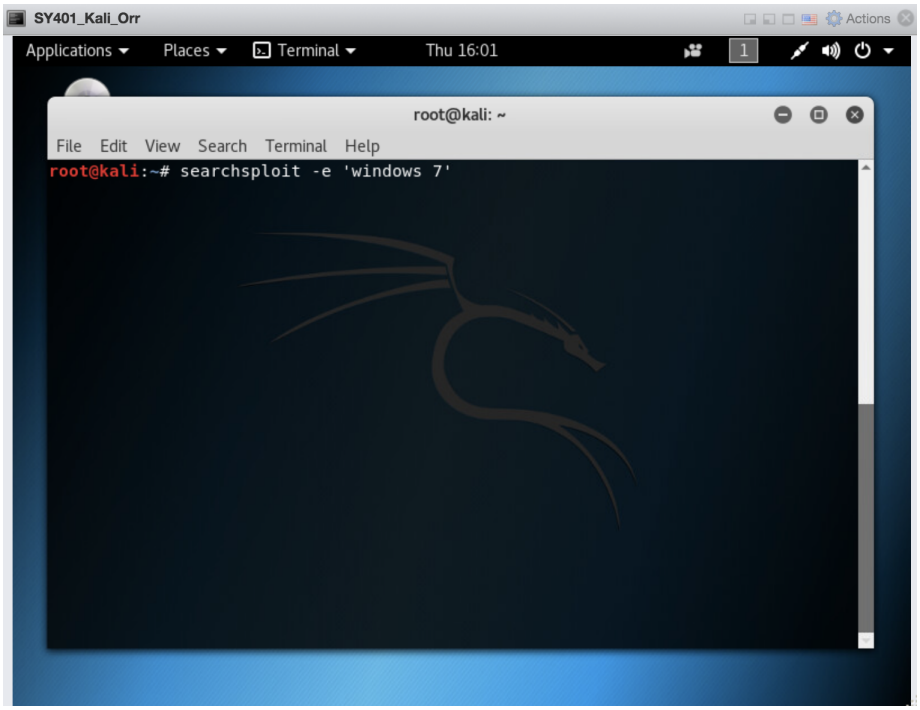
Cross-compiling exploits can be challenging. Exploit code is developed in varying environments and intended to work on a specific version and patch level of the software it is targeting. Many public exploits do not work straight out of the box. When cross-compiling, you will run across bad code, shell code that needs to be swapped, and other issues. With that out of the way, let's try out our new cross-compiler.

First, the exploit code itself. The best place to grab raw exploit code when using Kali Linux is the SearchSploit tool. Exploits can also be found on the web at exploit-db.com, securityfocus.com, and on many more sites. When compiling and running pre-written exploits, it is important that you trust the source or analyze the code yourself. There are plenty of malicious exploits out there!

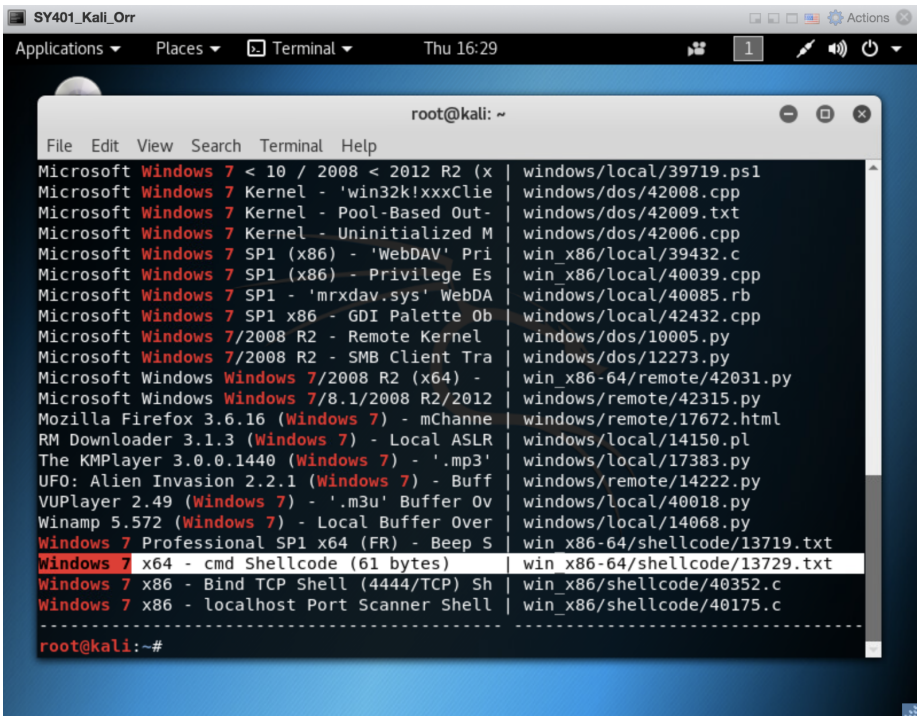
We will search using SearchSploit, by typing:

```
searchsploit -e 'windows 7'
```

The "e" argument specifies exact match.

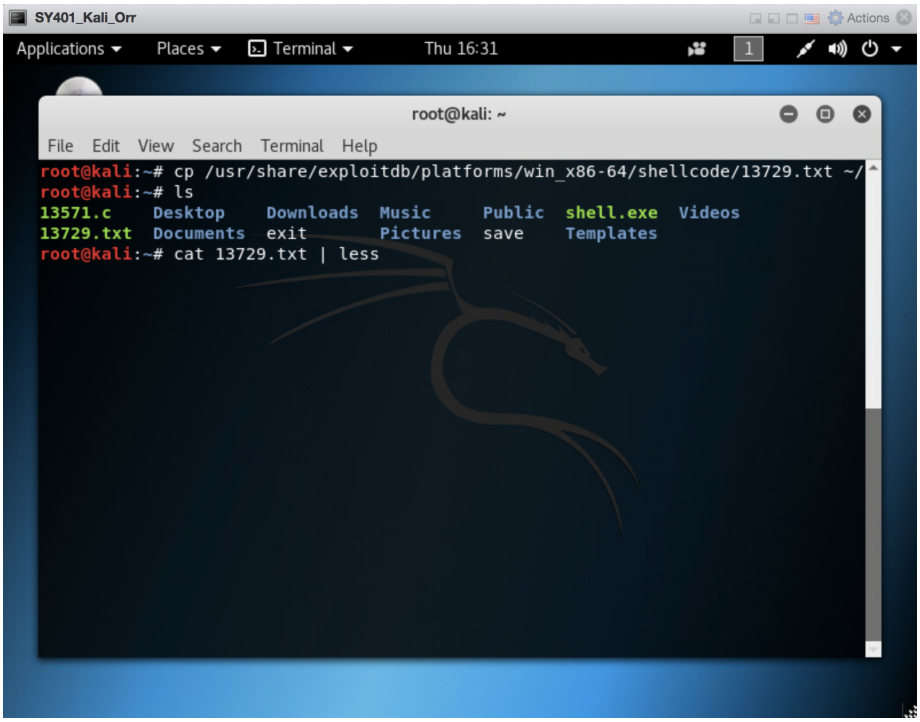


In the results from our search, we can see matching exploits. Though the highlighted result is labeled as a text file (.txt) by SearchSploit, it is, in fact, C code.



To copy the code into your directory, type the following command:

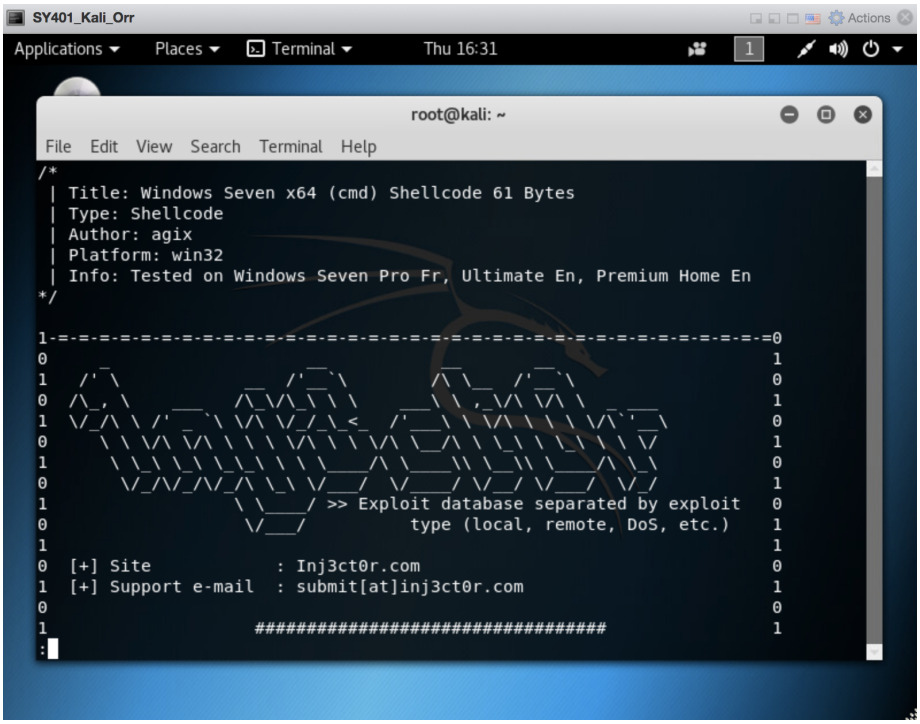
```
cp /usr/share/exploitdb/platforms/win_x86-64/shellcode/13729.txt ~/
```



Next, it's important to view the code before attempting compilation. In this example, we viewed the code with the command:

```
cat 13729.txt | less
```

The result is conveyed in the visual below.



Immediately, there is a problem. The banner in the code isn't commented out, which would lead to the compiler trying to read it, creating compilation errors. Good thing we checked.

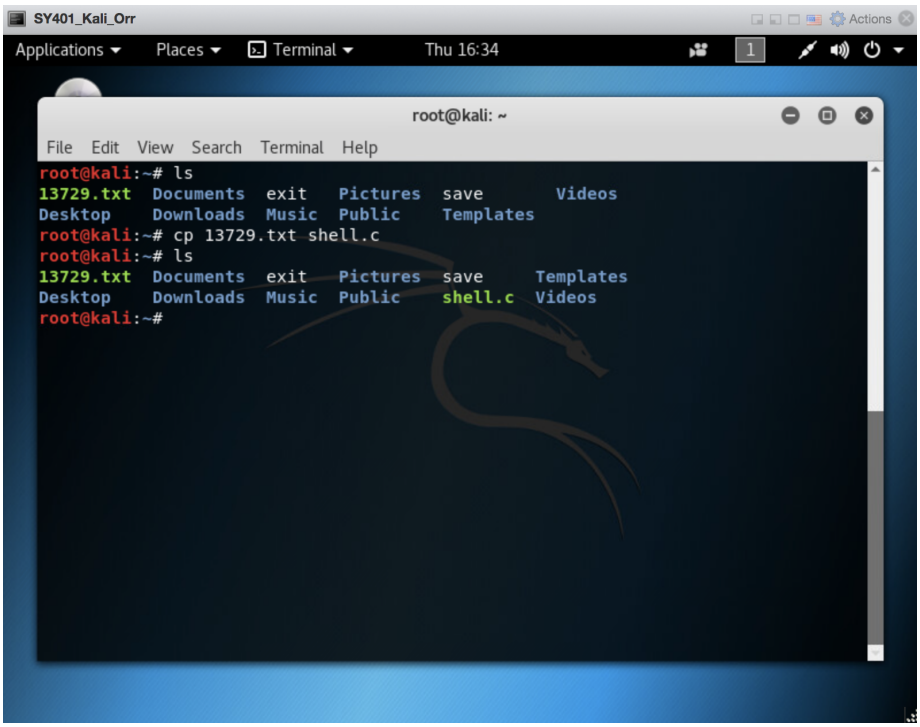
Beyond fixing that, there's nothing special happening here. This is just basic shell code. It would give us cmd.exe. If we wanted to spice it up a bit, we could remove the existing shell code and replace it with something generated by Msfvenom.

Since we know this code won't compile without removing the banner, we will make a copy of it to edit. Type the following command in terminal:

```
cp 13729.txt shell.c
```

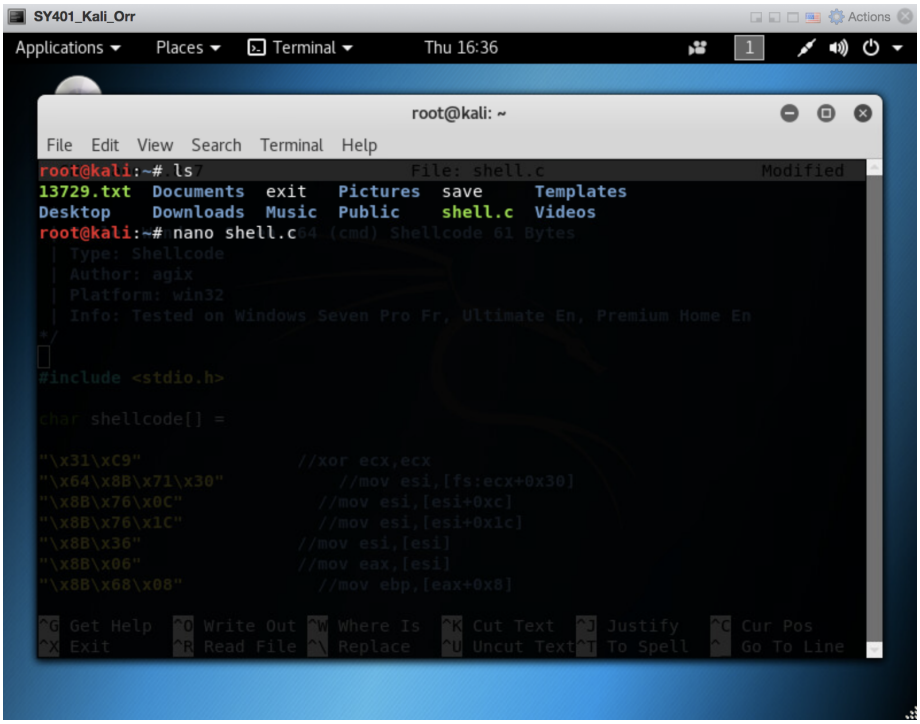
You may perform the following command to show the contents of the directory.

```
ls
```

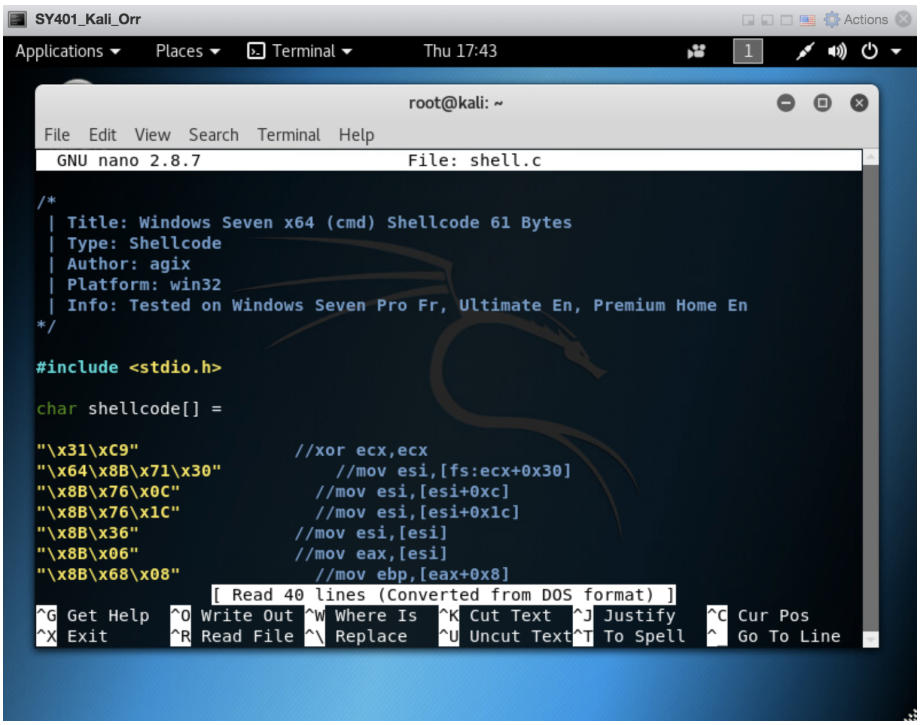


Next, open your favorite text editor and remove the uncommented text banner.

```
nano shell.c
```



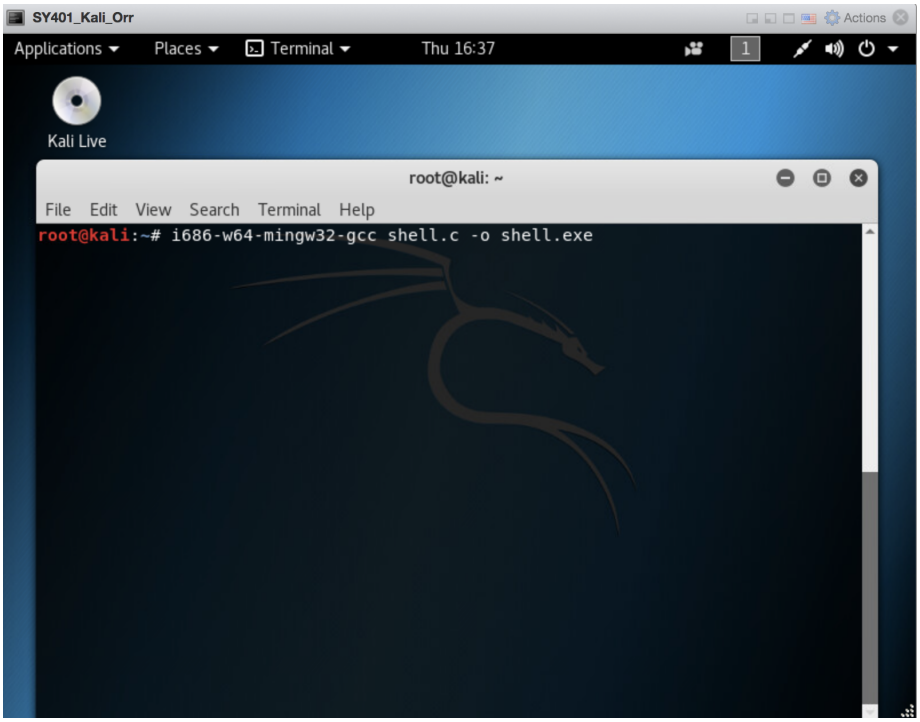
When you have finished, your code should look like the image below. In this example, we use the Nano editor.



The banner has been cleaned out, leaving behind around 40 lines of C.

To compile the code, run the command:

```
i686-w64-mingw32-gcc shell.c -o shell.exe
```



This command works for C files on x86 32-bit architecture. The "-o" determines the name of the compiled binary. If we were compiling for 64-bit, we would use a command that looks like this:

```
x86_64-w64-mingw32-gcc shell.c -o shell.exe
```

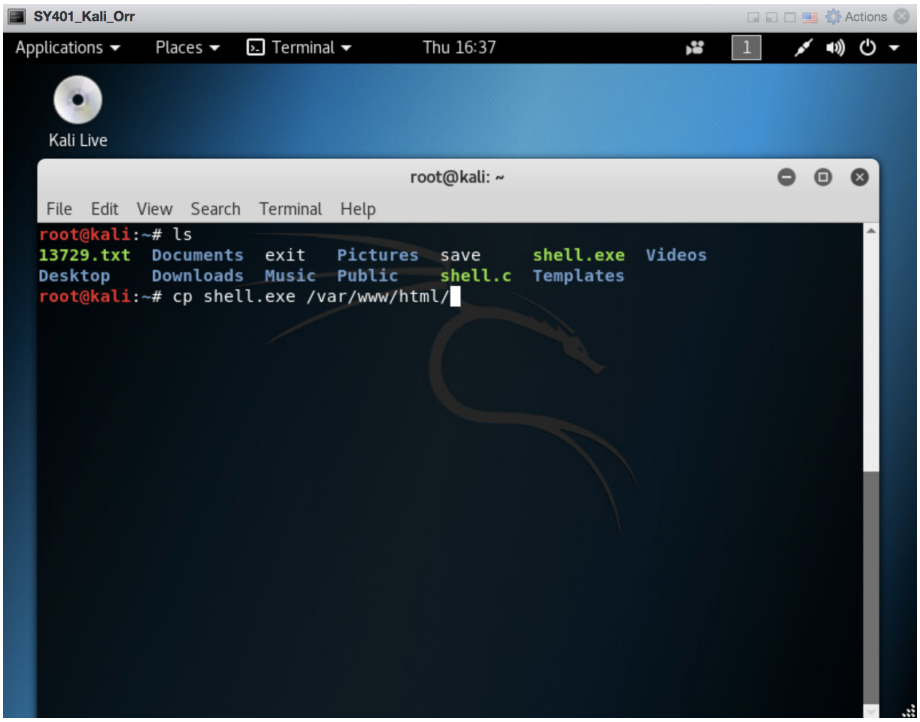
Since this code is for 32-bit machines (i.e., Windows XP), we use the 32-bit version of the cross compiler. MinGW comes with quite a few options for cross-compilation. In order to see a list of available compilers installed with the MinGW package, enter the following command:

```
apt-cache search mingw-w64
```

The final step is to run your compiled program on the Windows XP target VM.

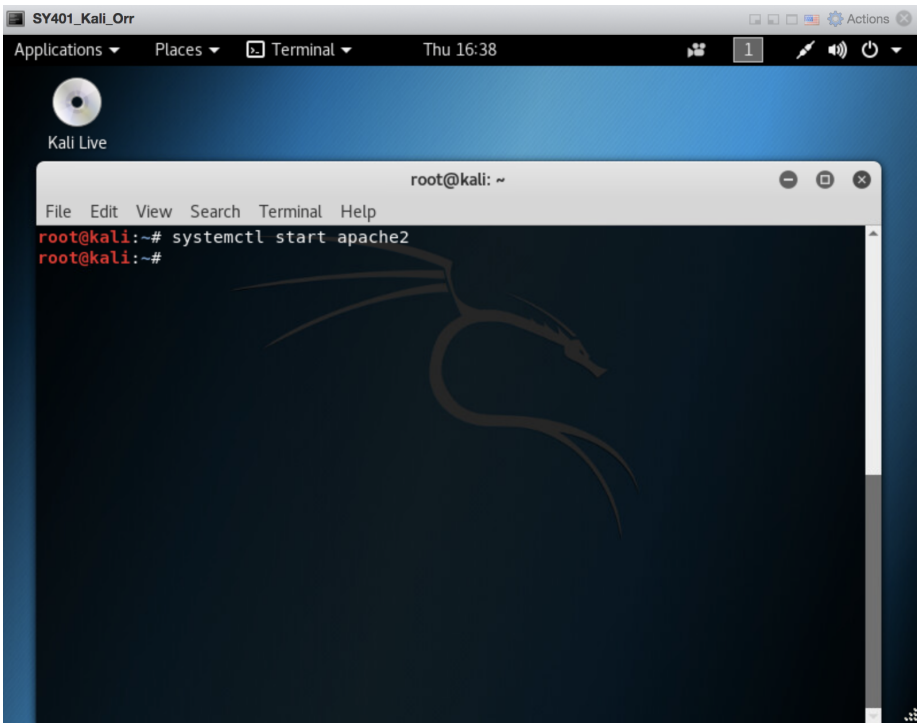
There are many ways to transfer files to your VM, but the easiest right now is simply by hosting the file on my Kali Linux Cyber Operation VM and pointing the target browser (i.e, IE) at it. In order to do this, run the command:

```
cp shell.exe /var/www/html/
```

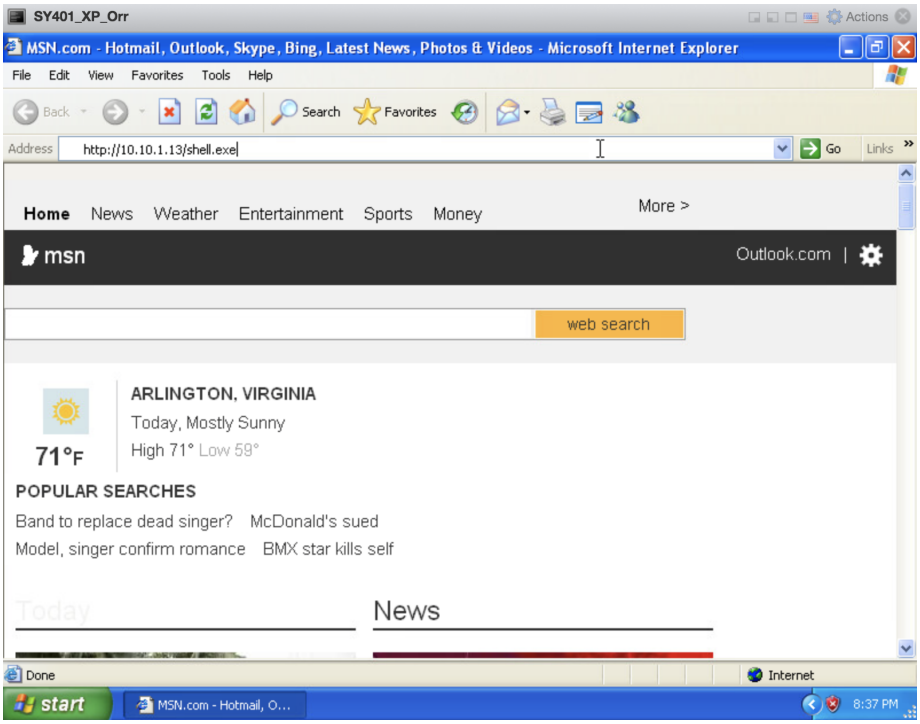


Now run:

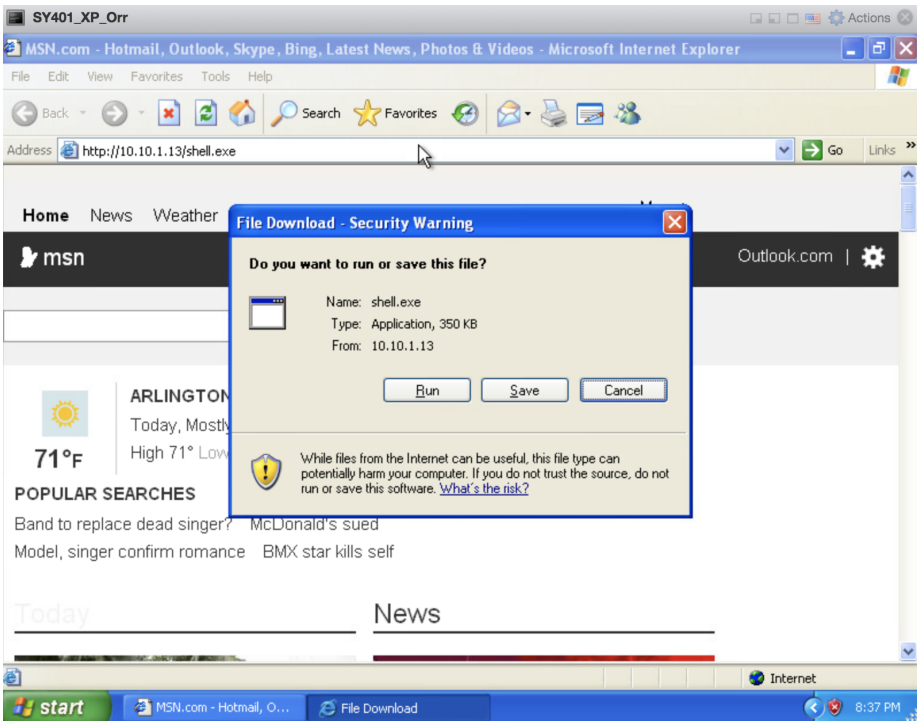
```
systemctl start apache2
```



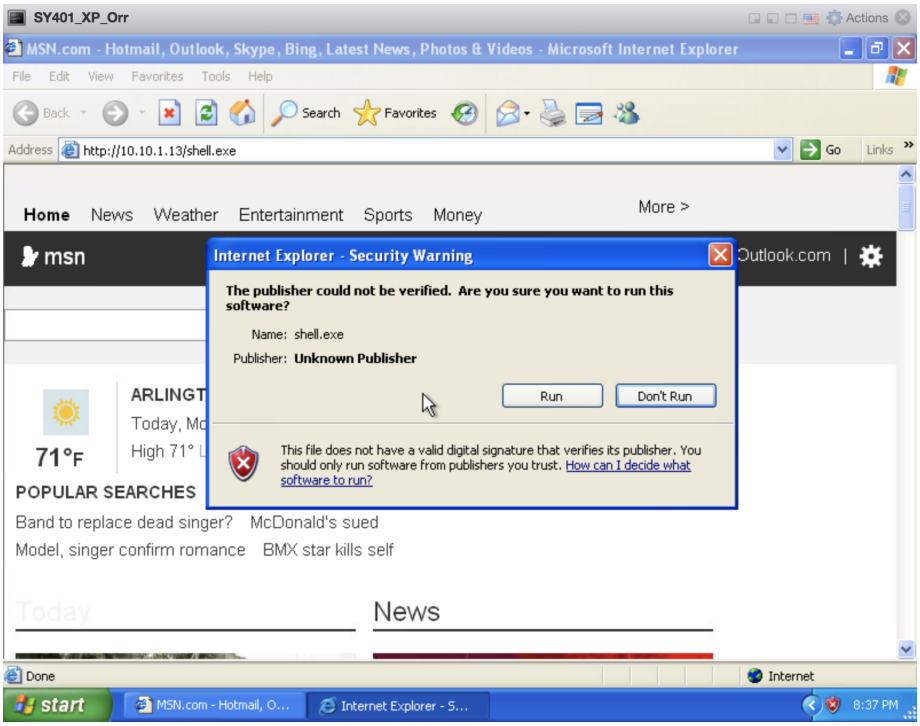
On your target system (i.e., Windows XP), navigate to http://your.kali.machine/shell.exe (with the "your.kali.machine" replaced by your Kali Linux VM IP address).



You will be presented with a prompt. Click on the save button.



Lastly, run the resulting "shell.exe" from a command prompt. When shell.exe is run, it immediately terminates, though we do see the cmd window open for a second. This could be adjusted by changing the shell code in the source and then recompiling it.



Congratulations! If you have successfully completed all of the steps above, and captured a screenshot on your system as is displayed in these instructions, you have achieved an 85%. Make sure to compile all of the screenshots into a PDF document and submit to your Professor.

For those of you wishing to push your knowledge and skills to the end, complete the remaining steps as discussed in the overview. That is:

- Research the exploit provided in the example or another of your choosing
- Modify the exploit source code
- Compile the exploit source code
- In addition to the above deliverable, provide a **screen capture** of your source code, and provide details of what you did/attempted
- If your exploit is unsuccessful in obtaining a shell on the host, describe what you think may be the issue**(+5%)**
- Take a screenshot of the Meterpreter shell once you have successfully exploited the Target VM with your modified code **(+10%)**

Good luck Cyber Operators!!

References

- (n.d.). Retrieved February 20, 2018, from http://www.ouah.org/readmeneu.htm
- Kapil, D. (2015, April 03). Buffer Overflow Exploit. Retrieved February 20, 2018, from https://dhavalkapil.com/blogs/Buffer-Overflow-Exploit/
- (n.d.). Retrieved February 20, 2018, from https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2016/june/writing-exploits-for-win32-systems-from-scratch/
- MinGW-w64 - for 32 and 64 bit Windows. (n.d.). Retrieved February 20, 2018, from https://sourceforge.net/projects/mingw-w64/
- Nano/Basics Guide. (n.d.). Retrieved February 20, 2018, from https://wiki.gentoo.org/wiki/Nano/Basics_Guide

- SearchSploit - The Manual. (n.d.). Retrieved February 20, 2018, from https://www.exploit-db.com/searchsploit/
- Offensive Security's Exploit Database Archive. (n.d.). Retrieved February 20, 2018, from https://www.exploit-db.com/
- SecurityFocus. (n.d.). Retrieved February 20, 2018, from https://www.securityfocus.com/