Lab 14: XLM macros

Table of Contents

Goals

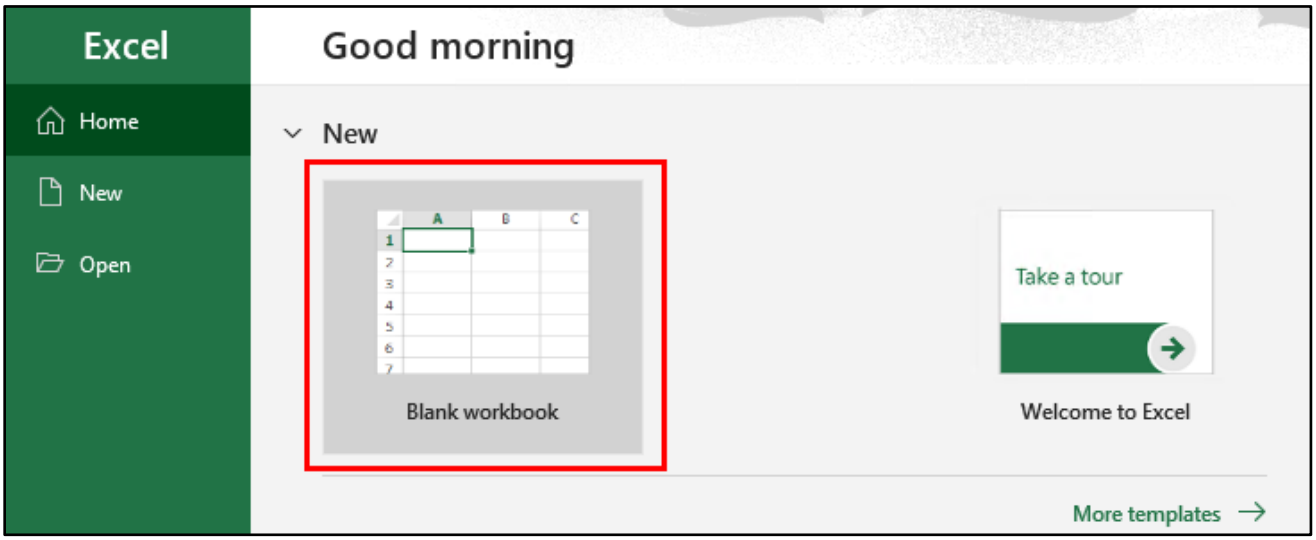
- Create an Excel spreadsheet that uses an XLM macro to execute a command when opened.
- Create an Excel spreadsheet that uses an XLM macro to execute shellcode when opened.

Requirements

- Windows 10 VM with Microsoft Office installed.
- Kali Linux Student VM.

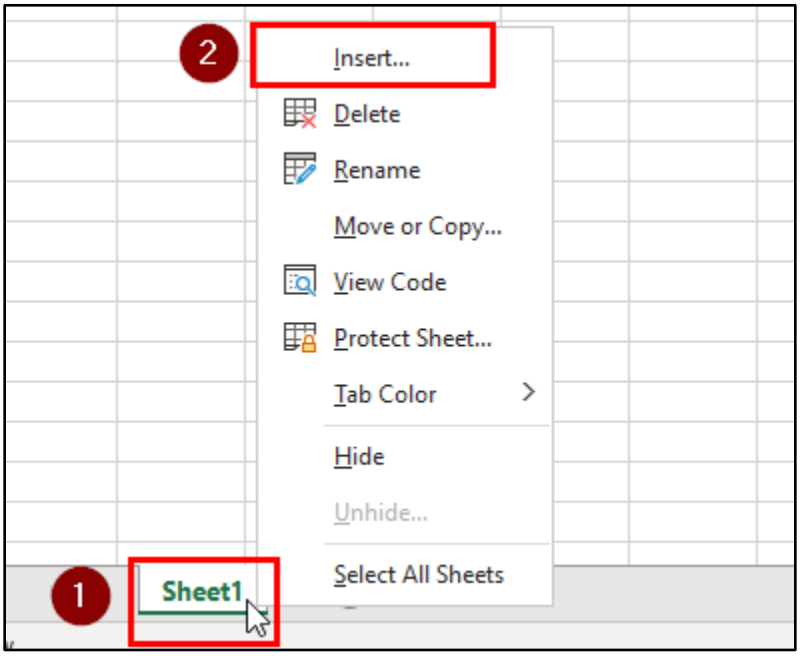
1. Create an XLM macro to execute a Windows command

1. In this stage of the exercise, you will create a simple XLM macro that executes a command. First, open Microsoft Excel and create a new spreadsheet document.



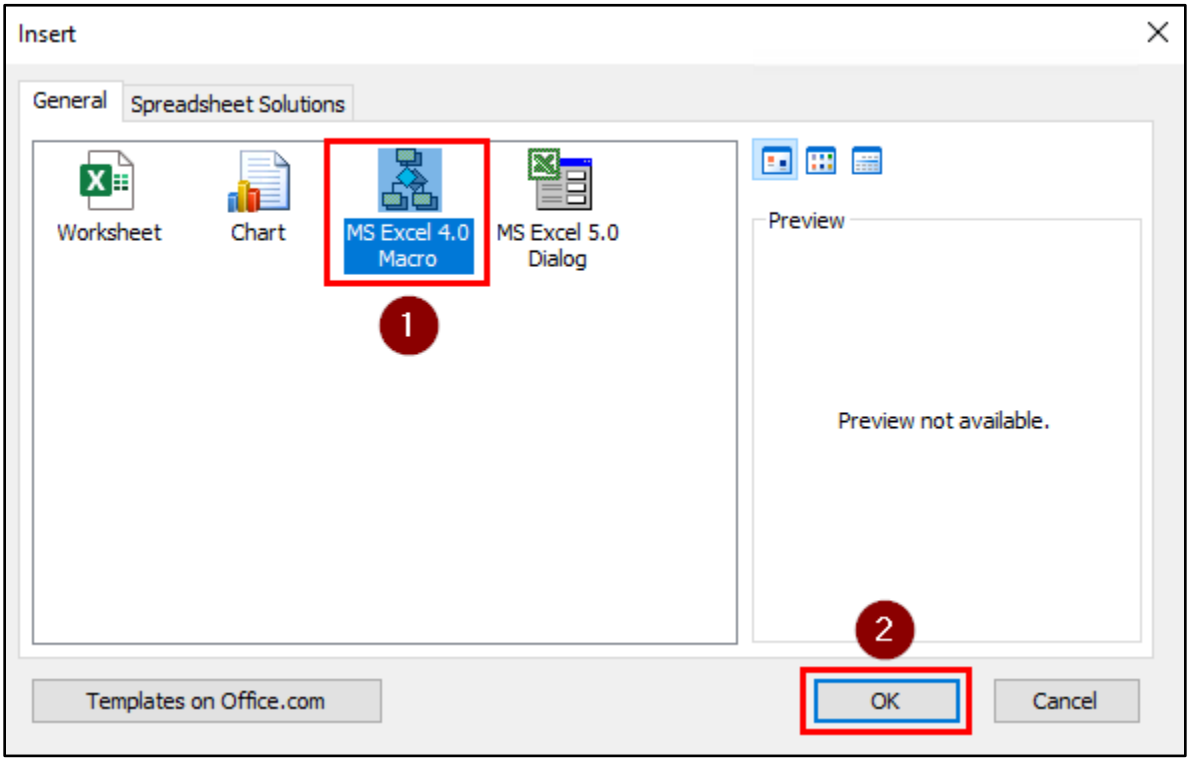
Creating a New Excel Spreadsheet

2. Next, right-click on the "Sheet1" tab at the bottom of the spreadsheet and then click "Insert..." in the menu that appears.



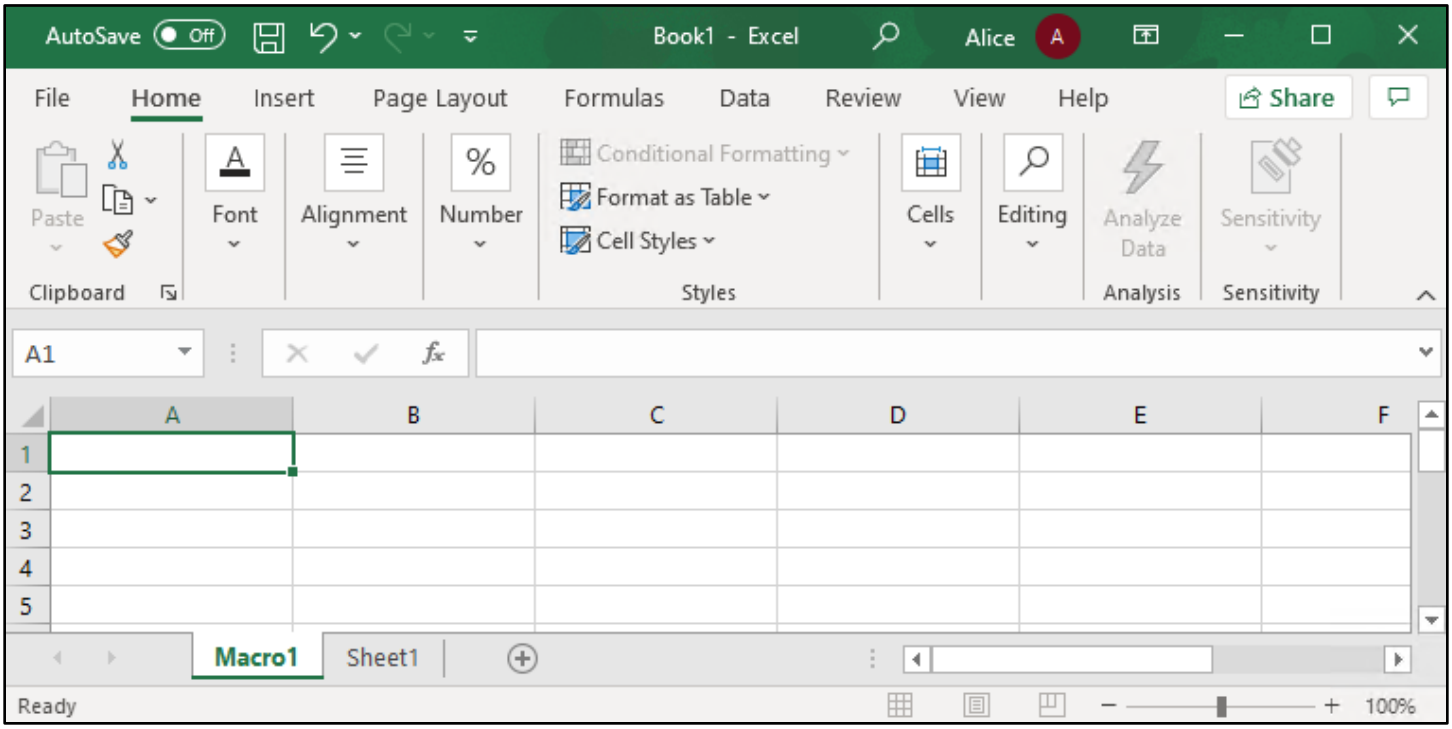
Clicking "Insert..."

3. In the "Insert" window that appears, choose "MS Excel 4.0 Macro", and then click OK.



Selecting "MS Excel 4.0 Macro"

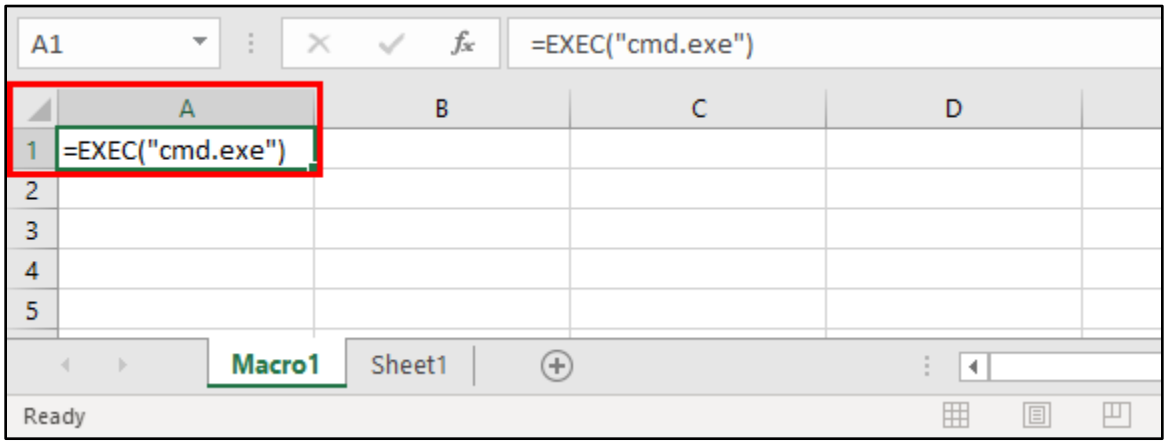
4. A new tab named "Macro1" should appear at the bottom of the spreadsheet.



Macro1 Tab Appears

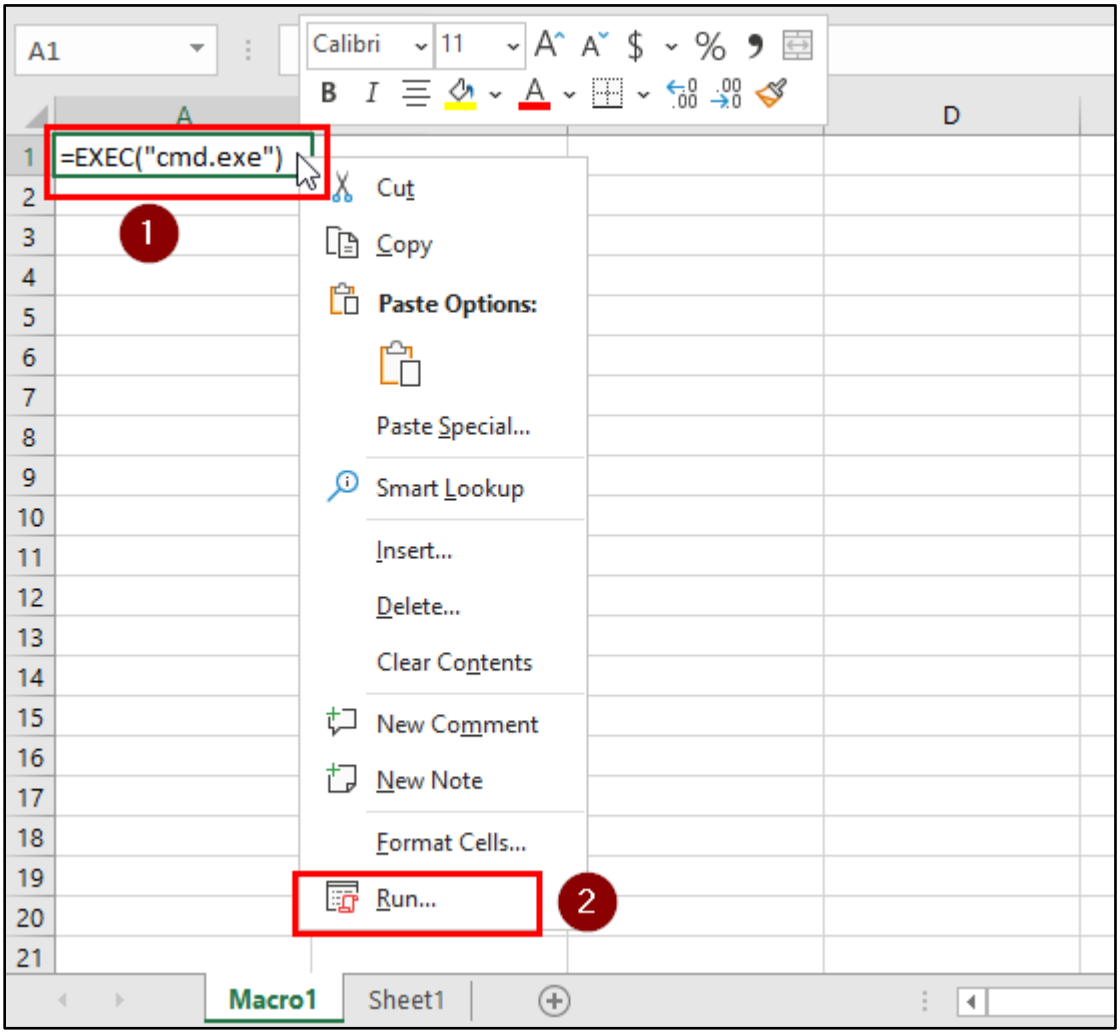
5. In the top-left cell (A1) of the Macro1 worksheet, enter the text shown below. This is a simple XLM macro that will execute the Windows Command Prompt when executed.

```
=EXEC("cmd.exe")
```



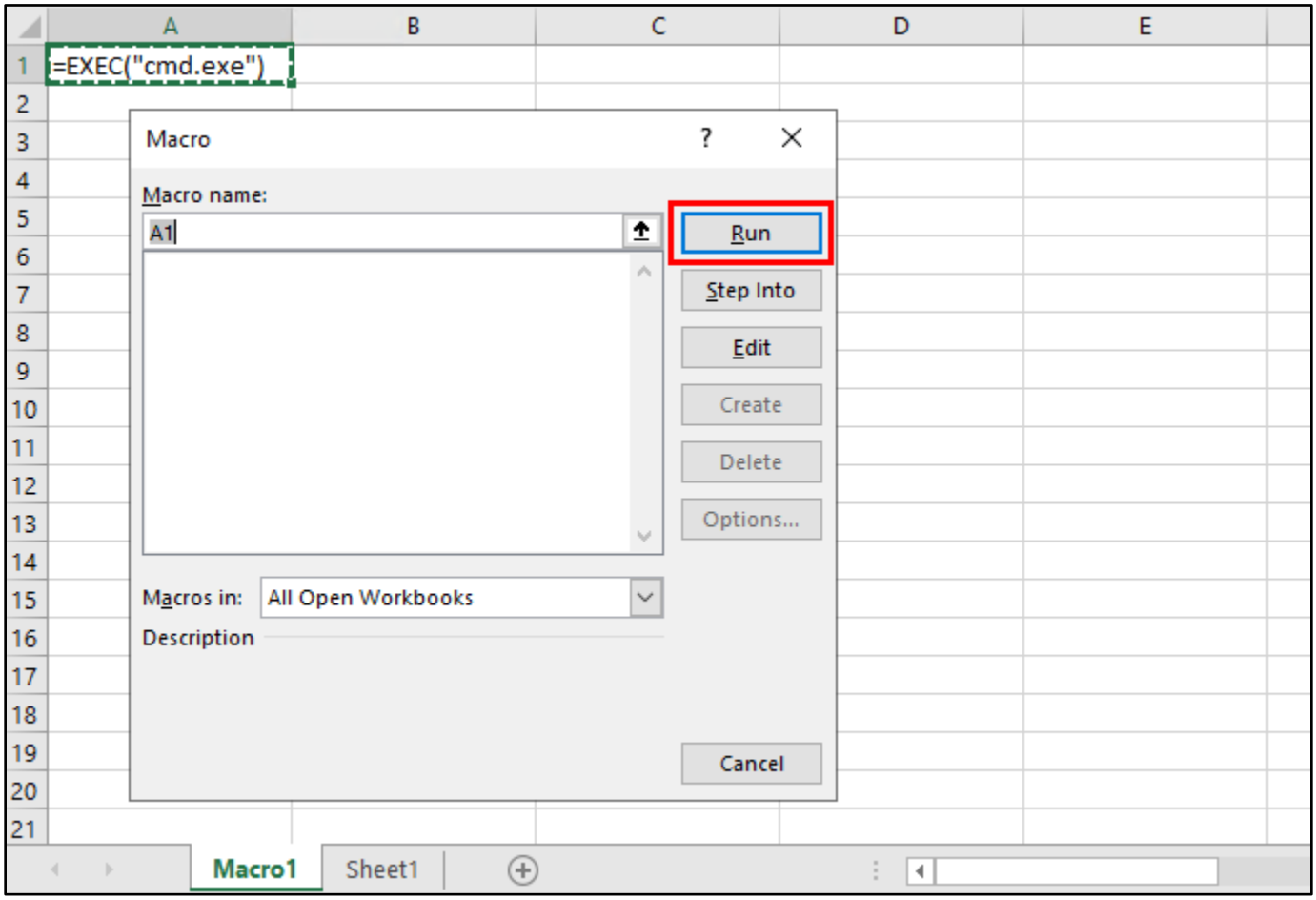
Simple XLM Macro

6. To test that the macro executes successfully, right-click on the cell and, then click on "Run…" in the menu.



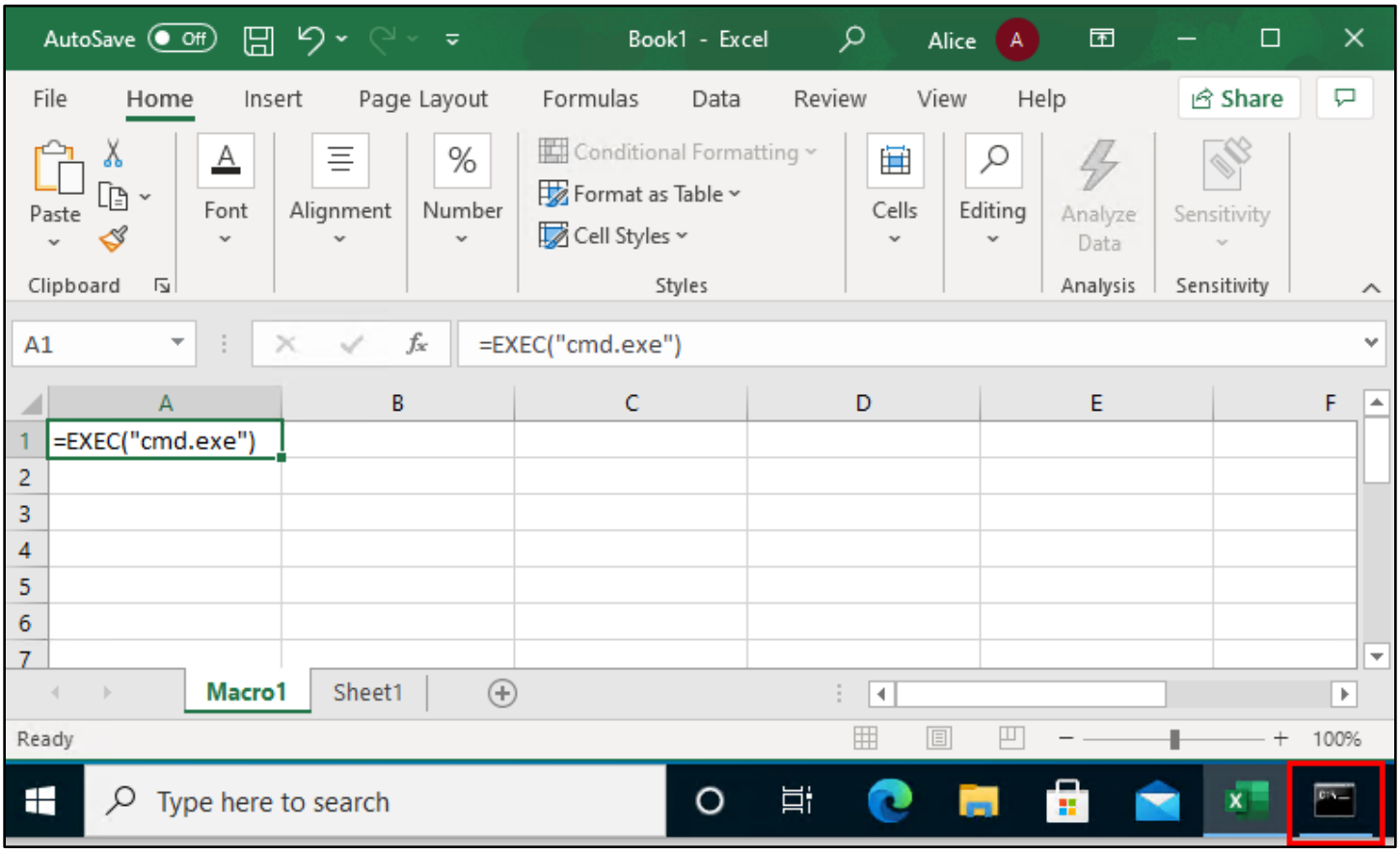
Testing Macro Execution

7. Then click on the "Run" button in the window that appears.



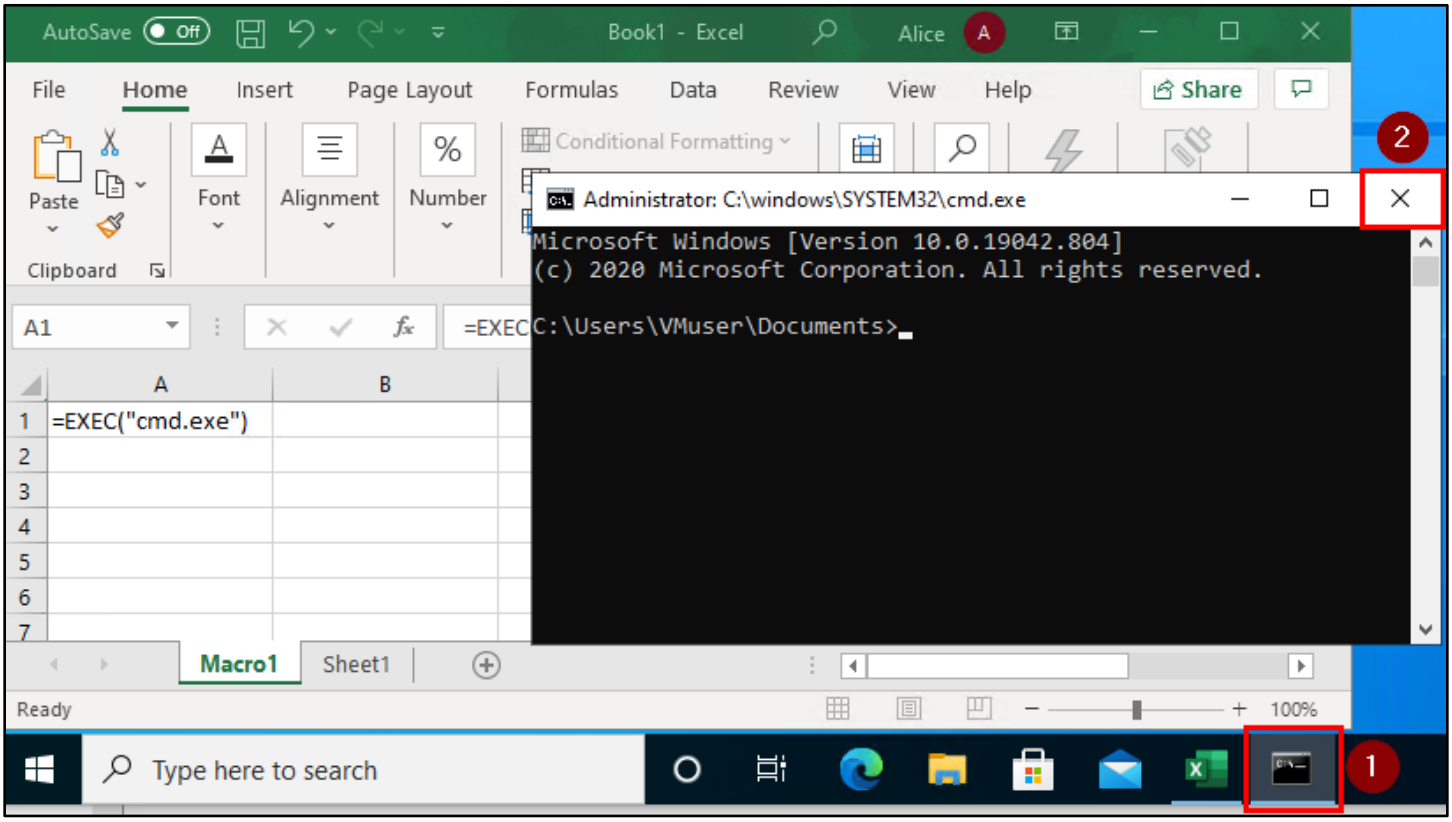
Clicking "Run"

8. Clicking the Run button should execute the "cmd.exe" command contained in your macro, which will open the Windows Command Prompt. If you don't see the command prompt window appear, check the taskbar - it may run in a minimized window.



Command Prompt Appears in the Task Bar

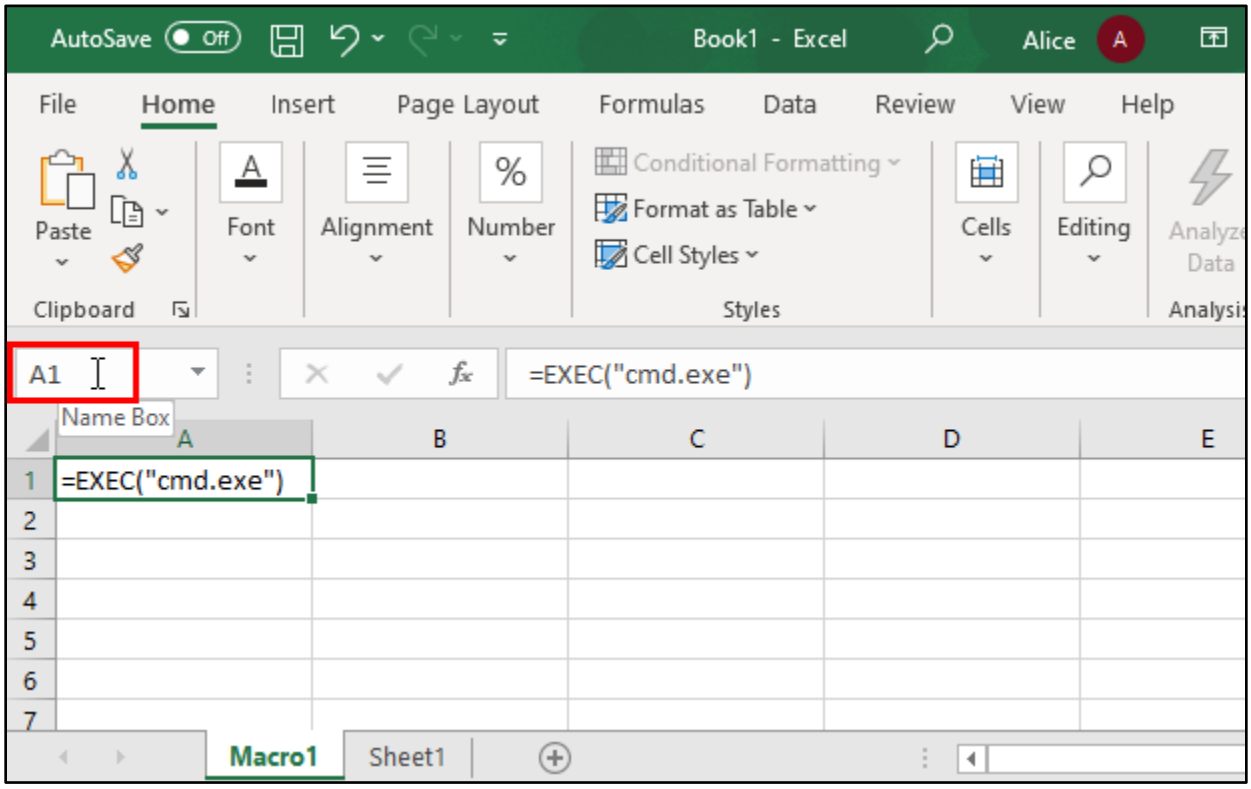
9. Click on the command prompt icon that appeared in your task bar, and close the command prompt window before continuing with the instructions in the next section.



Closing the Command Prompt

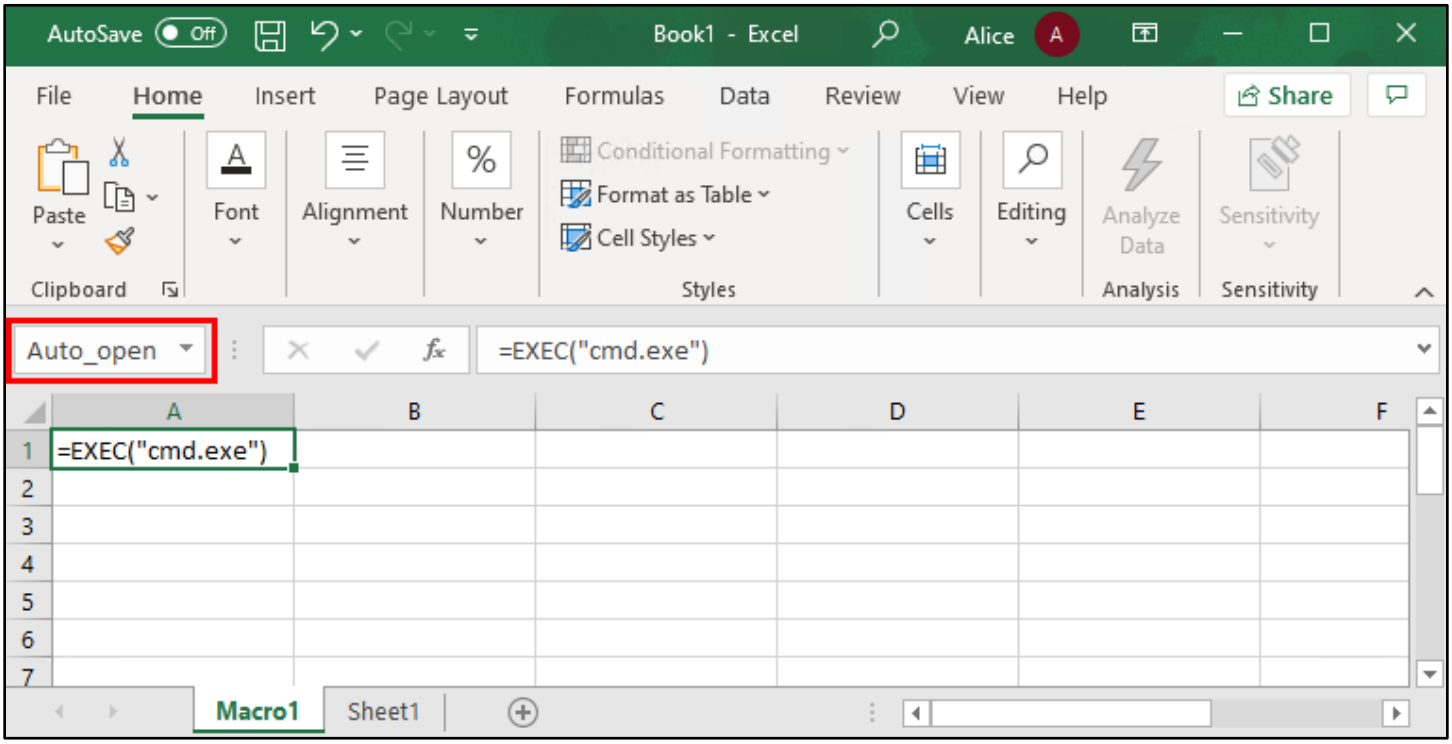
2. Making the macro automatically execute when the document is opened

1. Now you'll configure the macro to execute automatically whenever the spreadsheet is opened. Make sure that the cell where your macro is stored is selected, and then click in the Name Box in the toolbar above the cell.



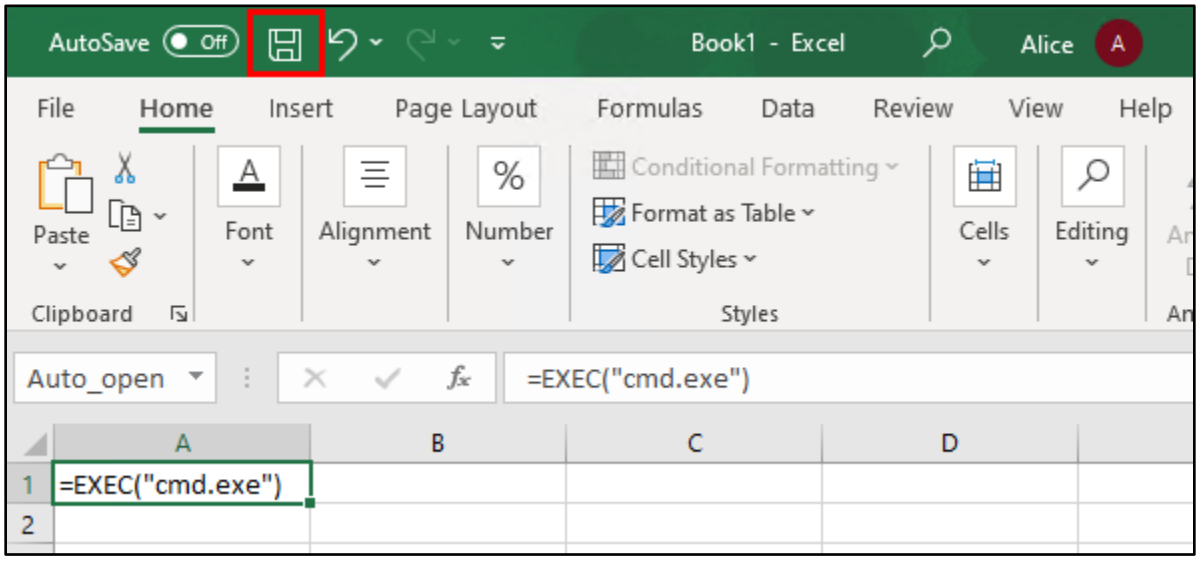
Changing the Cell Name

2. After you click inside the Name Box, change the name of the cell from "A1" to "Auto_open", and press Enter to confirm the change.

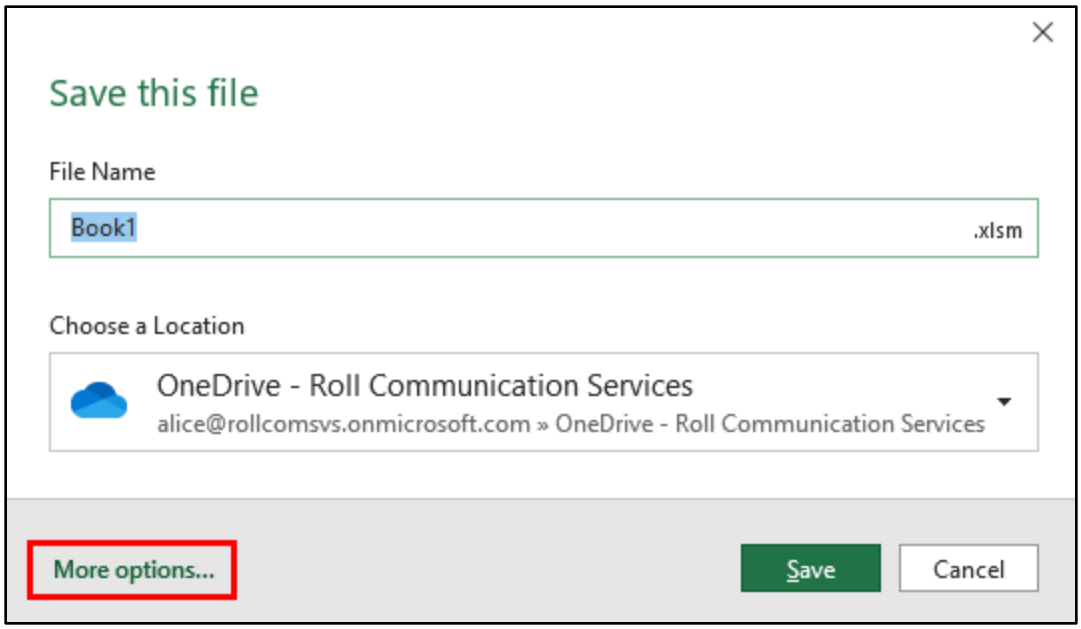


Name Changed to "Auto_open"

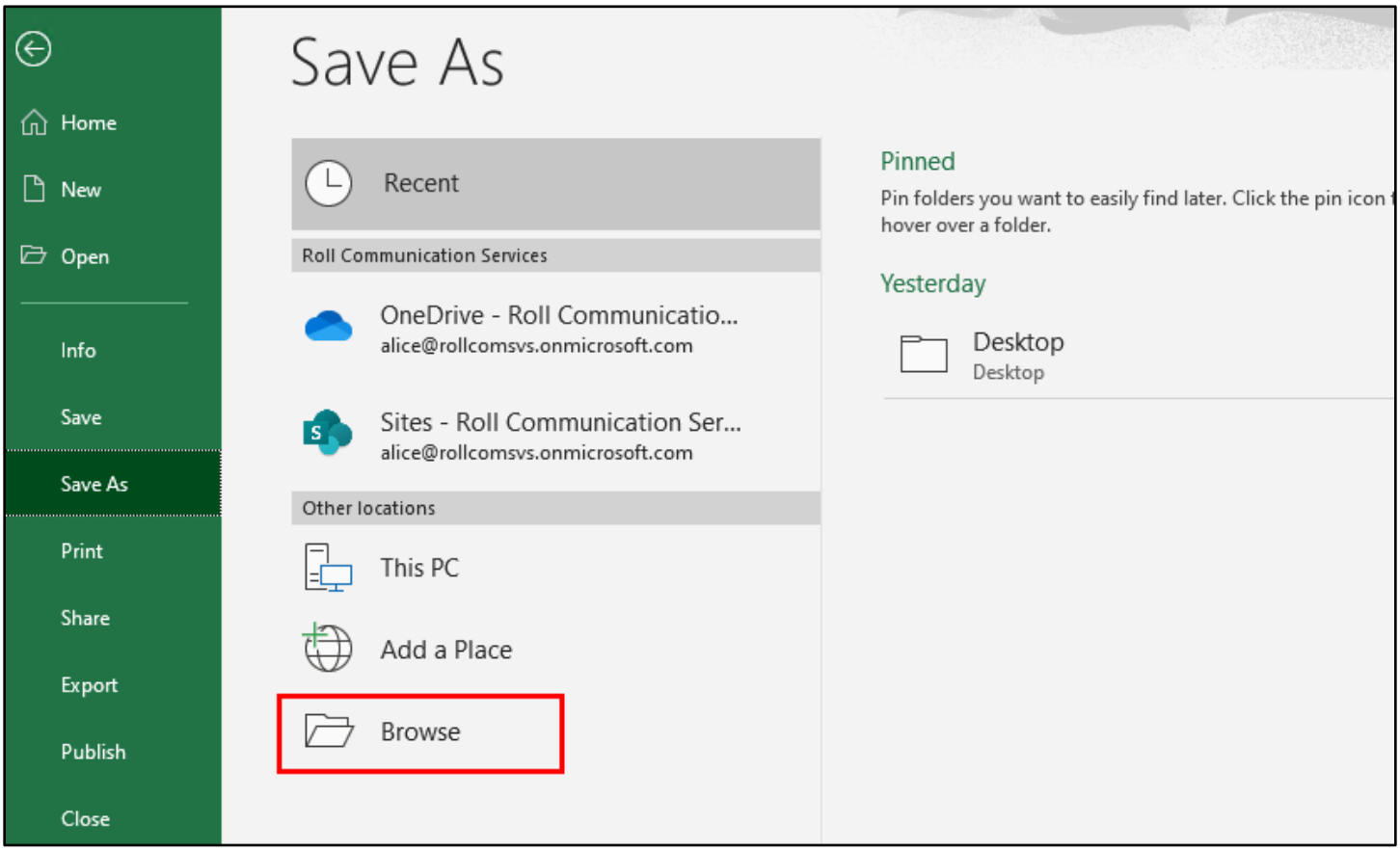
3. Now save the document by clicking the floppy disk icon at the top of the Excel window. Then save the file to your desktop as either an "Excel Macro-Enabled Workbook" (file extension .XLSM), or "Excel 97-2003 Workbook" (file extension .XLS). This process is illustrated in the four screenshots below.



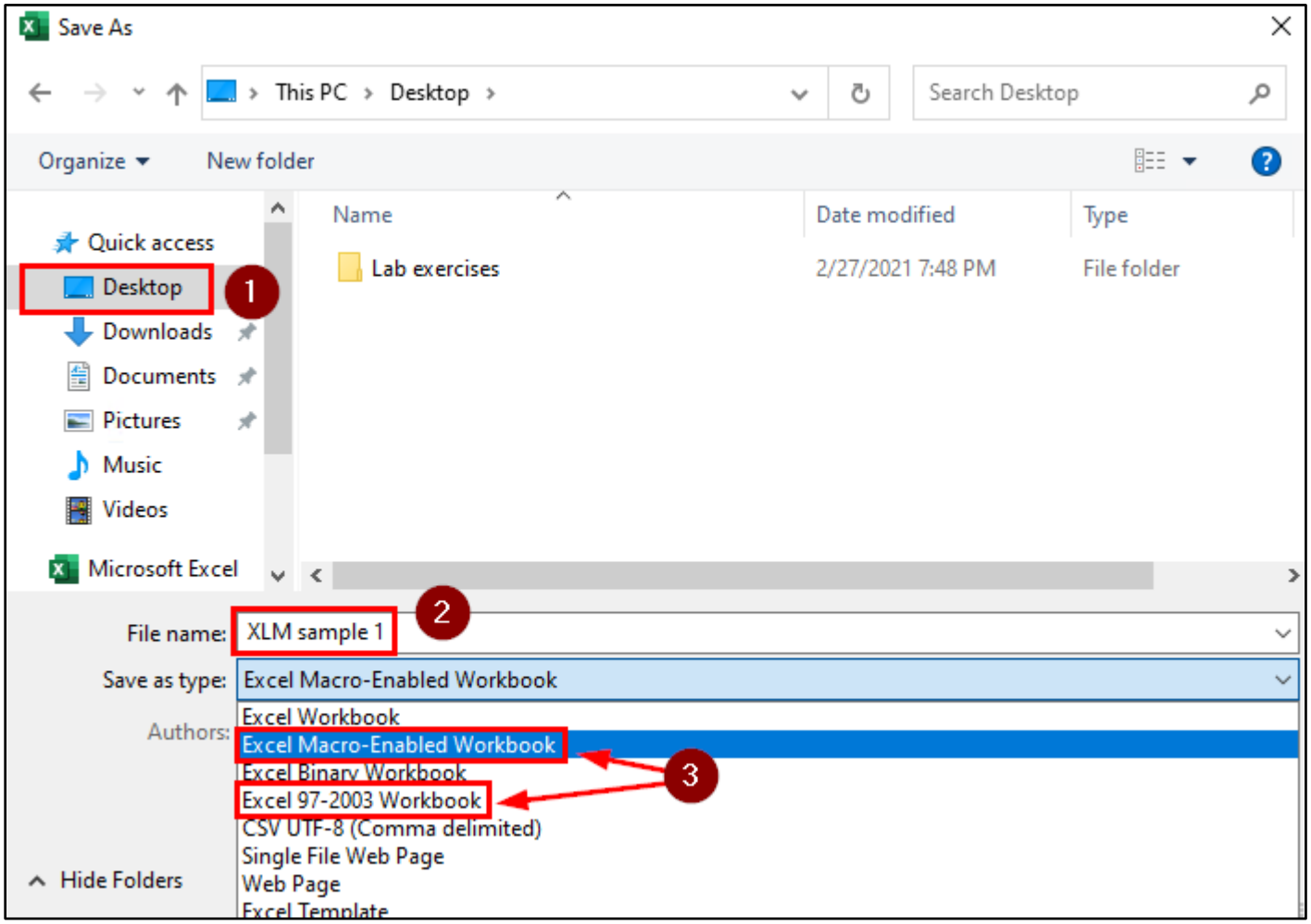
Clicking the Save Icon



Clicking "More options..."

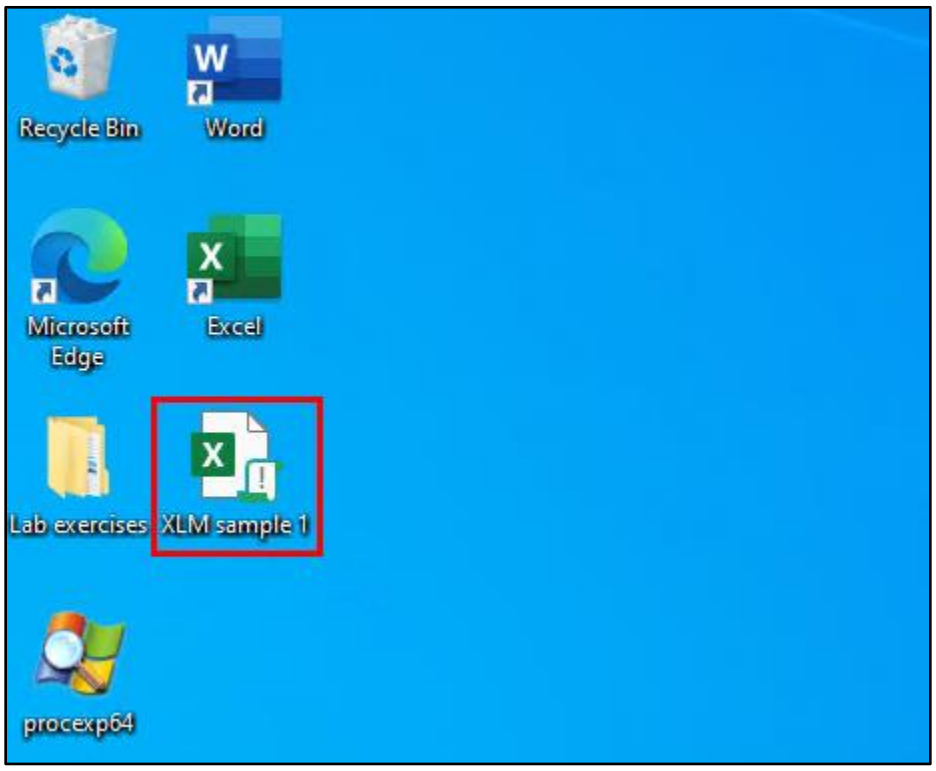


Clicking Browse



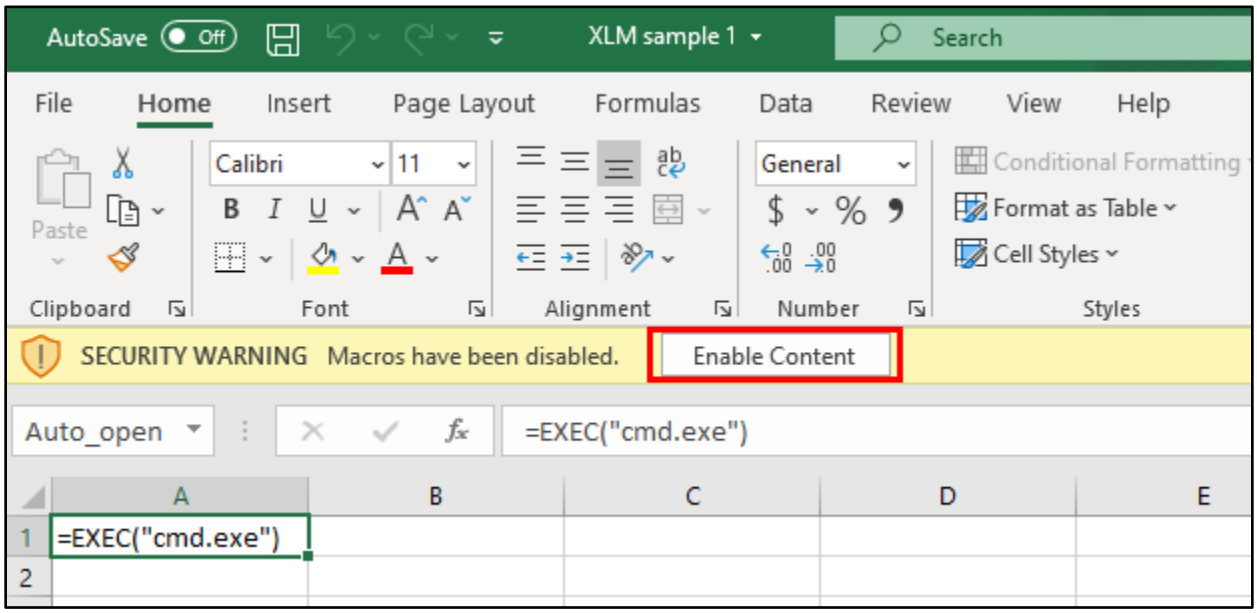
Saving as an Appropriate Format

4. After saving the document, close Microsoft Excel. You should see the spreadsheet file you just created saved on your desktop.



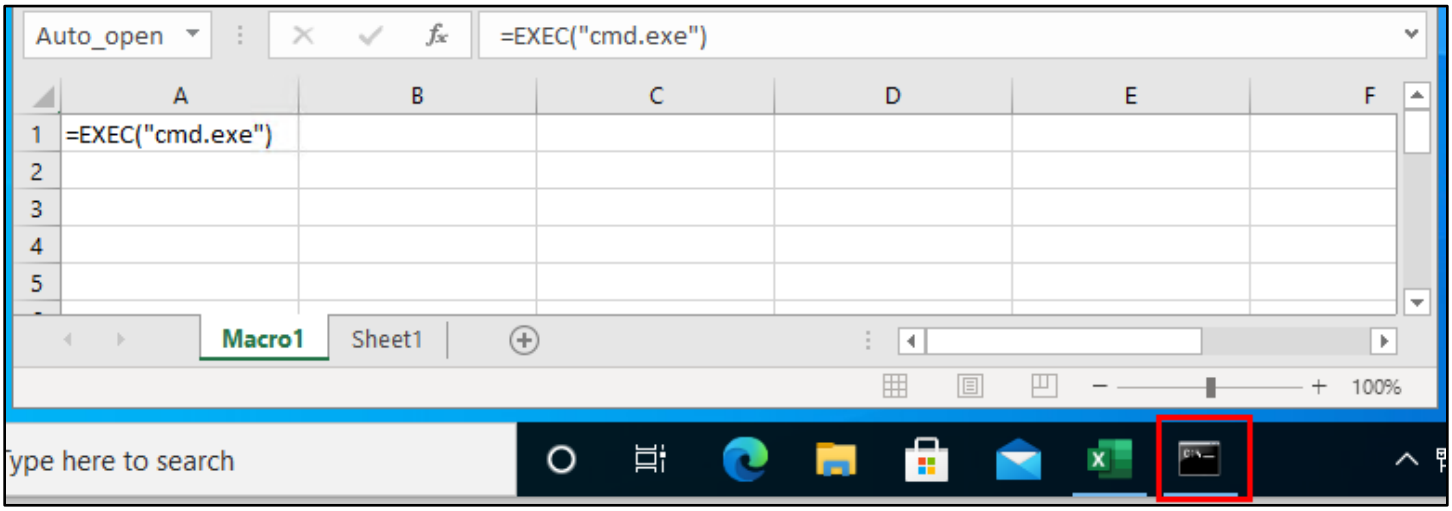
Excel Spreadsheet Saved to Desktop

5. Double-click on the file to open it in Excel again. When the file opens, you should see a Security Warning displayed at the top of the Excel window that indicates that the file contains macros. Click on the "Enable Content" button to execute the macro you created.



Clicking "Enable Content"

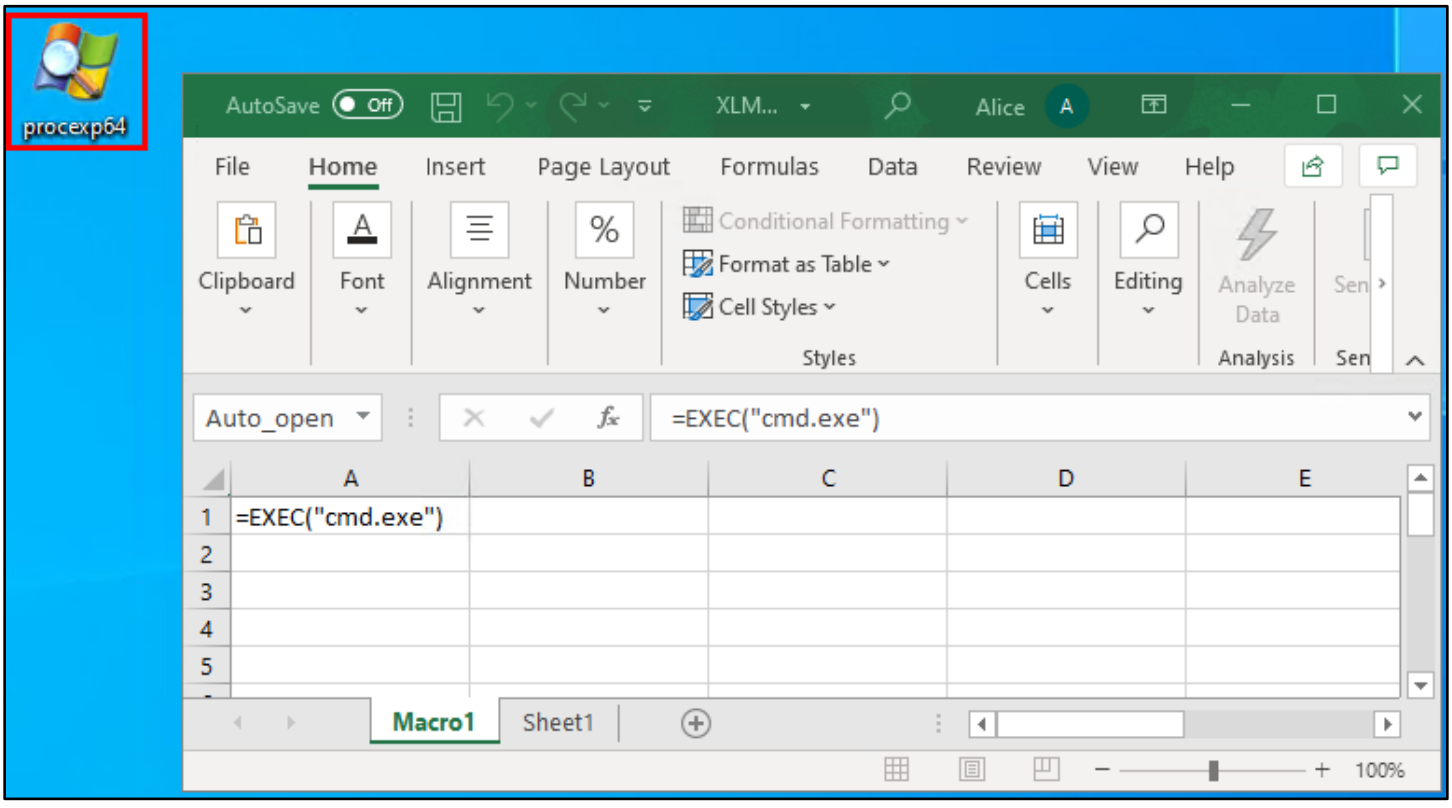
6. After you click "Enable Content", Windows Command Prompt should appear again. Remember to check the taskbar if you don't see the window open. This time, don't close the command prompt window after it opens. Leave it running for the instructions in the next section.



Command Prompt Opened in Taskbar

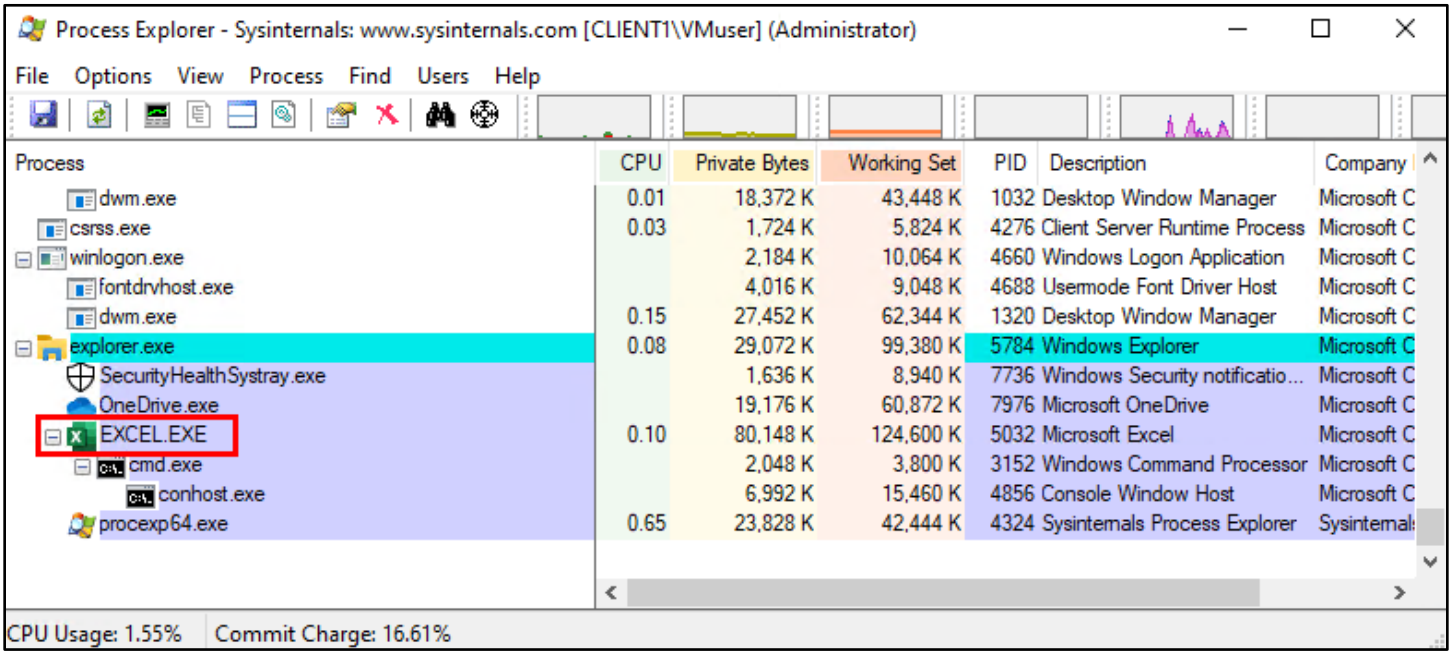
3. Observing the payload's process tree

1. With the Excel and Command Prompt programs still running, double-click on the "procexp64" on your desktop. This will open the Sysinternals "Process Explorer" program and allow you to inspect the process tree created your payload.



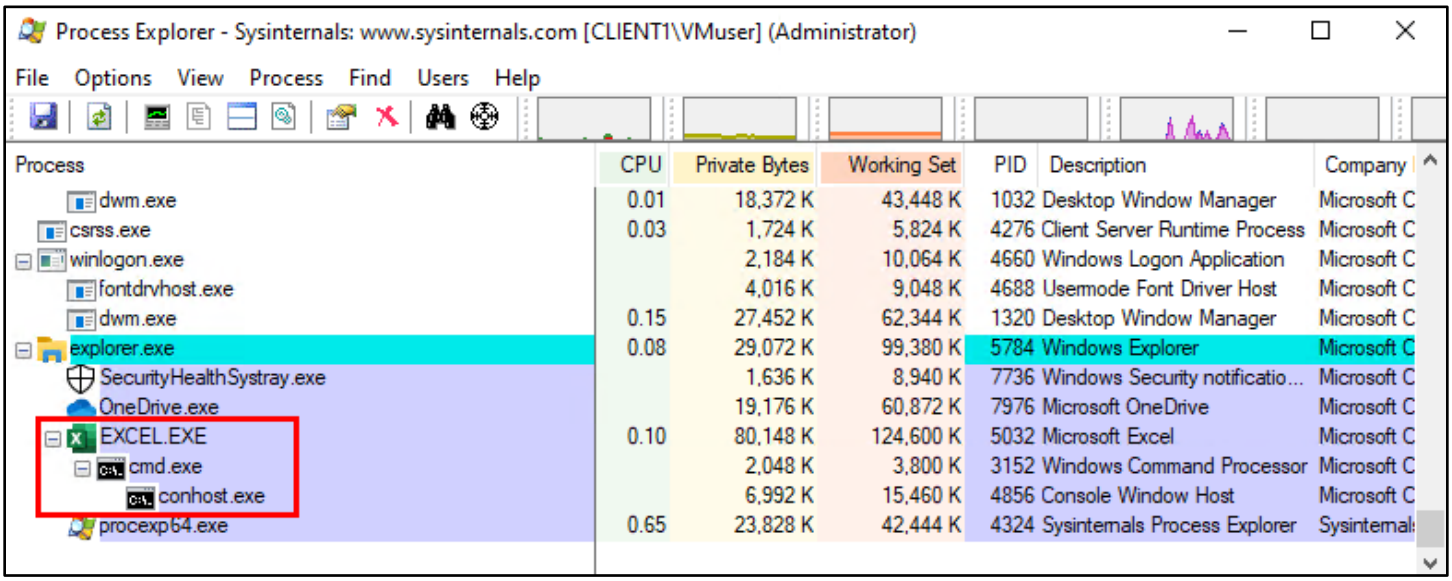
Opening Process Explorer

2. Within the Process Explorer window, scroll down until your find the Excel process, "EXCEL.EXE". It will probably be at or near the bottom of the list.



Finding EXCEL.exe

3. On the indented lines beneath Excel, you can see other programs that were started by Excel. This is known as the process tree - Excel is the parent process, and those processes branching down beneath it are its children. In this case, "cmd.exe" is shown beneath Excel. "conhost.exe" is also present, which was executed by "cmd.exe".



Excel's Process Tree

4. Since Excel doesn't execute "cmd.exe" or "conhost.exe" during normal use, this is suspicious behavior that may be detected by defenders. Furthermore, console programs like "cmd.exe", "powershell.exe", and "conhost.exe" (which gets spawned by both cmd.exe and powershell.exe) are **even more suspicious** because they are frequently executed by attackers but less frequently executed by the majority of end users.

   In other words, "cmd.exe" being a child process of Excel (like it is here) would be a very strong indicator that an attack is taking place.

