Delivery Mechanisms:

**Office Documents**

# Malware Activity

**Lab Description:** We’re going to be working with **oledump** from Didier Stevens. You can download **oledump.py** from:

<http://blog.didierstevens.com/programs/oledump-py/>

**Lab Environment:** Use of oledump is needed for this lab. It is recommended to do this lab in a **REMnux** virtualized environment. The file being provided could be malicious. You may need to install additional dependencies to get the script to work. Once you have the script working, review the documentation by invoking the script.

**$ python oledump.py -m**

You may want to pipe it through more to allow you to read all the output:

**$ python oledump.py -m | more**

**Lab Files that are Needed:**

Samples.zip

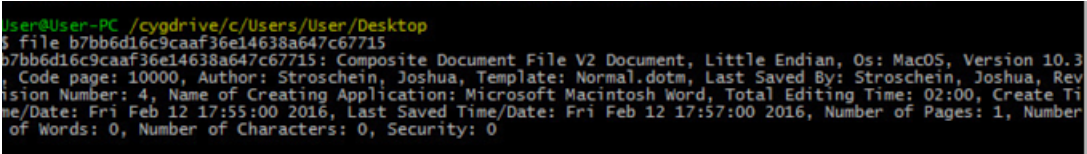
### **Lab Exercise 1**

*Learning Outcomes 1 & 2*

Using the following file, answer the questions below.

**MD5: b7bb6d16c9caaf36e14638a647c67715**

* Use the file utility from a Cygwin terminal to inspect the file:



* Use Oledump to inspect the file, does it contain macros? If so, inspect them.
* Is this file malicious?

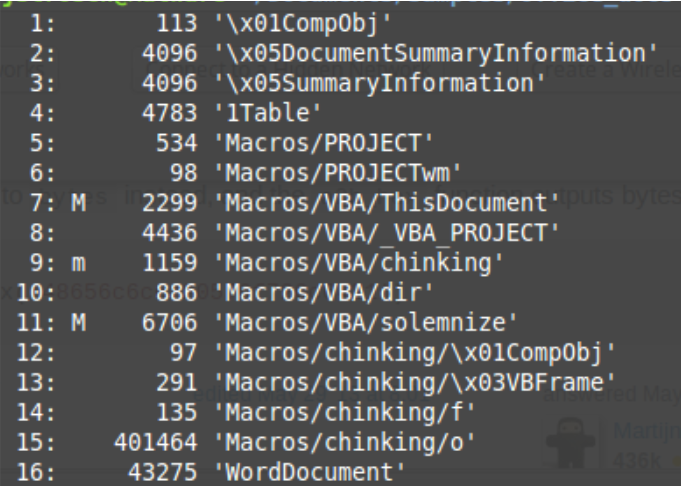
### **Lab Exercise 2**

*Learning Outcomes 1, 2, & 3*

OLE files can be viewed as an archive containing **streams** and **storage**. Streams will appear as embedded data within an OLE container. Storage will appear as folders and can contain other streams and storages. Perform the following tasks:

**MD5: 1b0fbd5e0af361058a8115b941232e34**

* View the streams, which streams contain macros?  Your output should look like the following:



* Extract and inspect the macros, what function is used to execute the macros?
* Do any other streams look interesting/worth inspecting? Why or why not?
* Is this file malicious?

### **Lab Exercise 3**

*Learning Outcomes 1, 2, & 3*

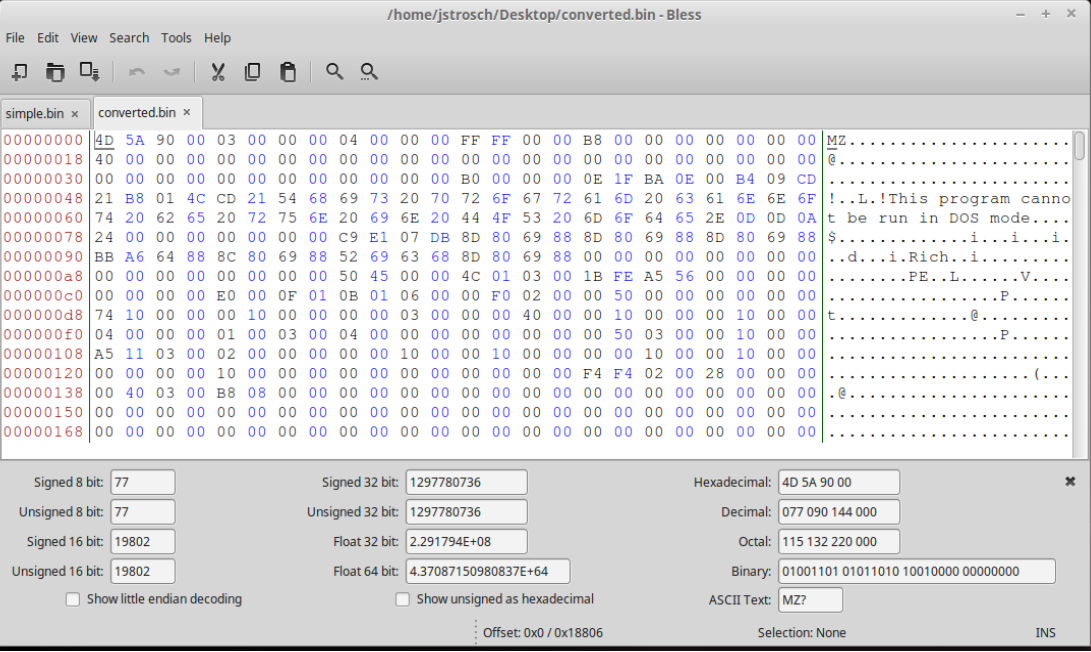
Using the following file, answer the questions below.

**MD5: 78989bc304554d5dfe1bbea3794cd0b**

* What file type is this sample (ie Word, PowerPoint, etc)?
* The macro code makes a connection to a URL, what is the URL?  Use static analysis only!
  + HINT: You won’t find it as ASCII text, you will need to convert integer values into chars.  I wrote a small (~ 5 lines) Python script to do the work for me.

### **Lab Exercise 4**

File 2 (1b0fbd5e0af361058a8115b941232e34) contains an embedded EXE, find the executable and extract it - confirm that you have a PE file by analyzing it (run **file** on it, upload to VirusTotal, etc). I wrote a small Python script to help me ‘convert’ the embedded content into the EXE. Once you’ve done it correctly you should see something like the following in a HEX editor:



# What to submit

Submissions should be neatly organized. Each question should include at least one screenshot and a brief explanation if possible.