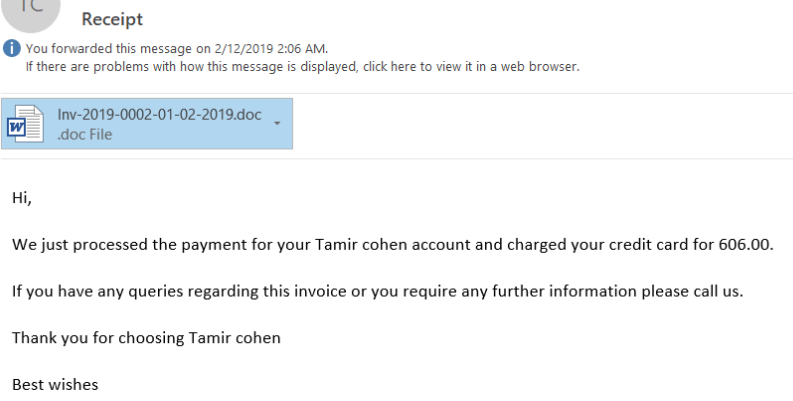
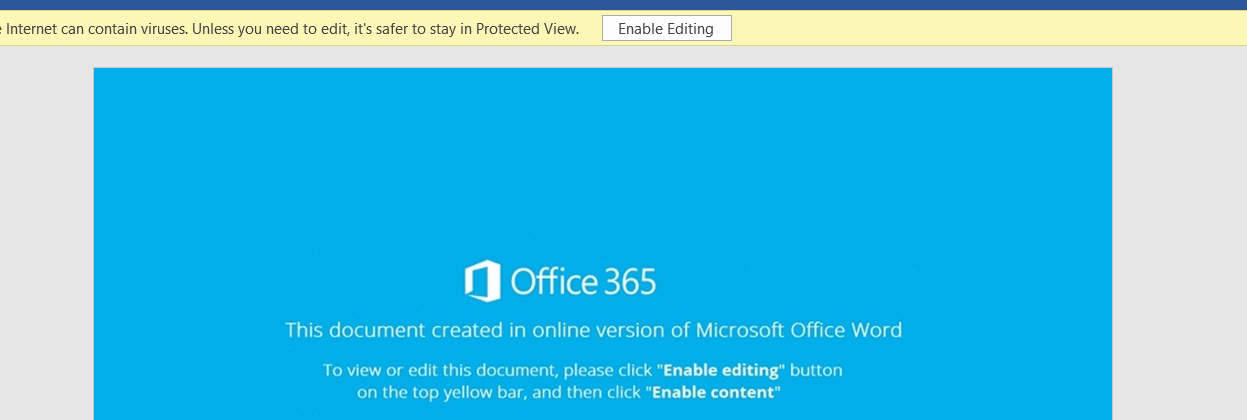
**Analyzing Phishing email.**

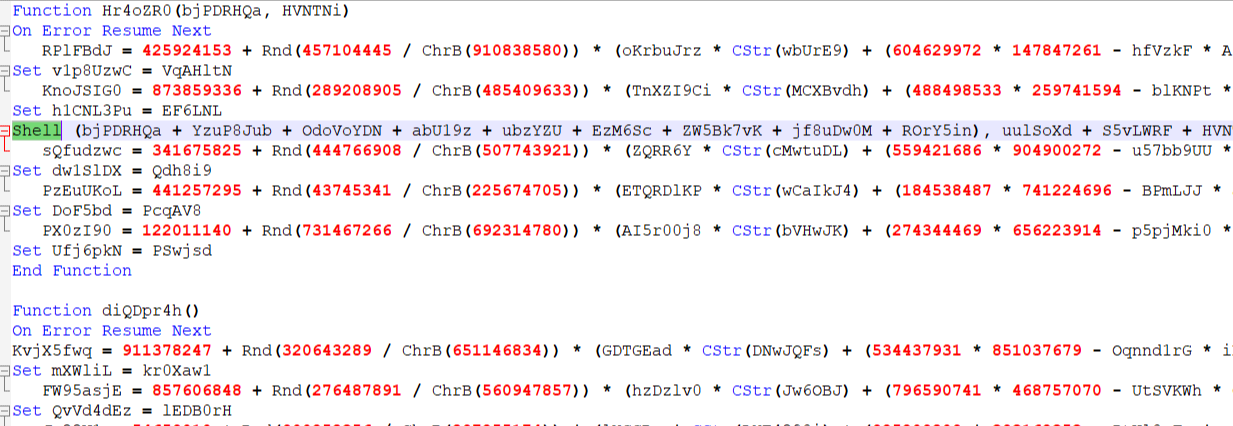
In 2019 cyber attacks becomes more sophisticated and harder to detect. Most organization these days getting infected from external sources such as emails attachments, visiting compromise websites, clicking on infected links and more. The attached word document I have checked was sent to someone in organization with malicious macro payload inside. The employee was very careful about and didn’t open the document.



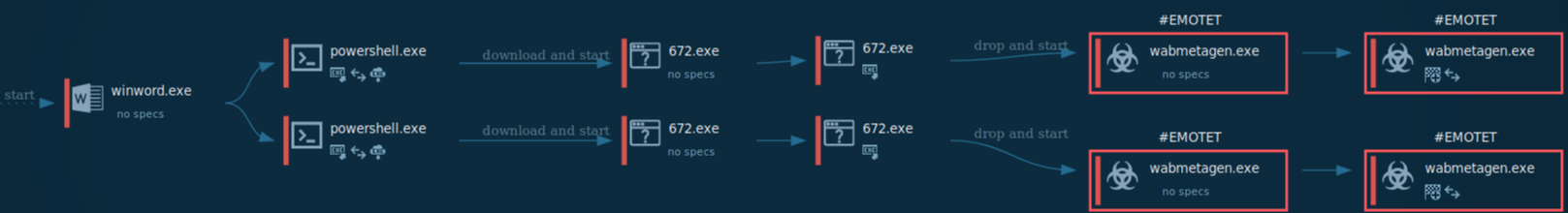
The first step was checking the document in safe environment, to see what is doing. Basically, when the document is opened you must click “enable editing” to view the content, immediately after that the malicious macro code executing background commands through PowerShell.



The PowerShell code creates process and downloading the second stage of the malware in other word the document is a dropper for malware. From developer point of view, I was curious to see the code and to understand the full attack vector. Doing the analysis I looked on the macro code however it was obfuscated with some hint word ‘Shell’.



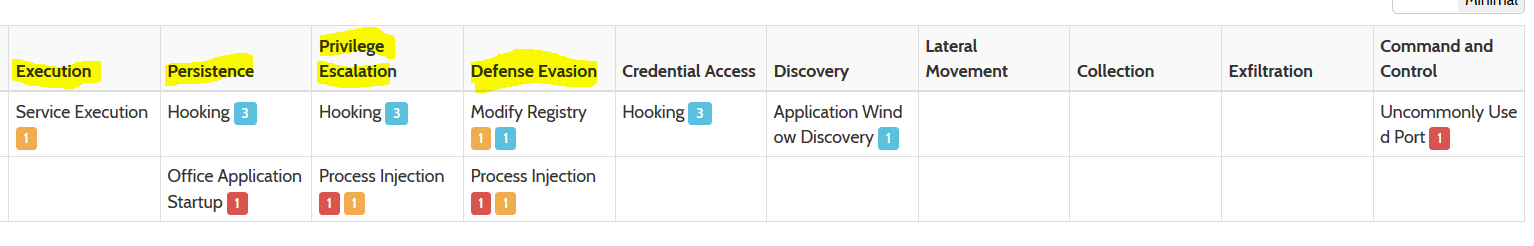
From sandbox analysis I knew the basic malware flow, but I won’t able to see the code inside the word document:



HTTP requests:

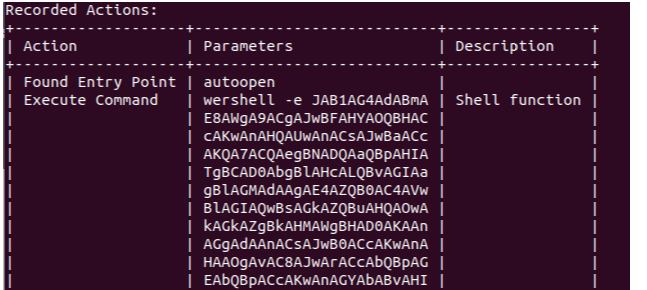


The dropper file matched to the MITRE ATT&CK techniques:



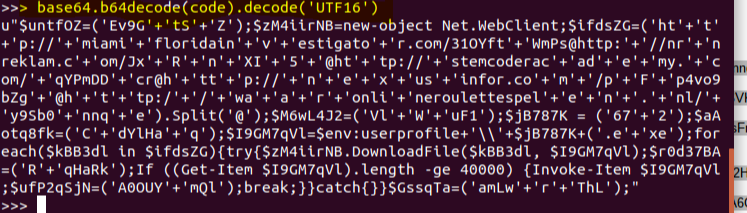
So, the dropper looks juice, to view the macro I used viper monkey tool simple python script for macro analysis.

The usage of a tool is simple -> python vmonkey -s ‘filename’



Look like I have the payload, is it base64 encoded command?

I wrote three lines of python to decode the payload an image attached below.



Then the output above is more readable, in many programing languages the character ‘+’ used to concatenate a string, so I formatted little bit the output by deleting unwanted characters. After I found the variables and changed them to some meaningful names.

The original code looked something like this:



Reference to online tools first response:

<https://any.run/report/872e1bdbf5efcd65c8280f1c916940efe191d41b65e71613b9c4417ef333cea1/47f29acb-07c9-4385-b8b8-c17a6de037a2>

<https://www.hybrid-analysis.com/sample/dc890cdbf81c9a5e6bce33592ad1a527ec2a49d368771901f3ab21dc7114c7e3?environmentId=120>

Wrote by**: Alex .**